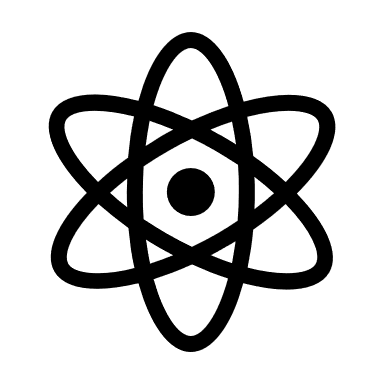
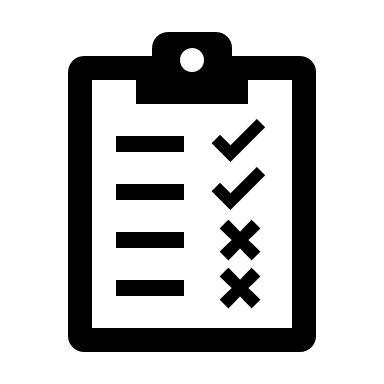
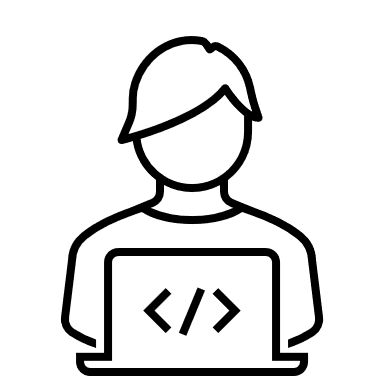
**3D Portfolio Website**

Programmieren eines Portfolios

Ein Bild, das Text, Computer, Multimedia, Elektronik enthält.

Automatisch generierte Beschreibung

********

|  |  |
| --- | --- |
| Autor: | Jann Fanzun |
| Lehrfirma: | soxes GmbH |
| Abgabetermin: | 28.06.2023 |

Inhaltsverzeichnis

[1 Änderungstabelle 4](#_Toc138774250)

[2 Projektidee 5](#_Toc138774251)

[2.1 Projektauftrag 5](#_Toc138774252)

[2.2 Ausgangslage 5](#_Toc138774253)

[2.3 Ziele 5](#_Toc138774254)

[3 Informieren 7](#_Toc138774255)

[3.1 Planungstechnologien 7](#_Toc138774256)

[3.2 Entwicklungsumgebung 7](#_Toc138774257)

[3.3 Frontend 8](#_Toc138774258)

[3.4 Deployment Methoden 8](#_Toc138774259)

[4 Planen 9](#_Toc138774260)

[4.1 Arbeitspakete 9](#_Toc138774261)

[4.2 Gantt-Diagramm 10](#_Toc138774262)

[4.3 Kanban Board 11](#_Toc138774263)

[4.4 Wireframe 12](#_Toc138774264)

[4.4.1 Laptop 16:9 12](#_Toc138774265)

[4.4.2 Telefon 19.5:9 13](#_Toc138774266)

[4.5 Flow Chart 14](#_Toc138774267)

[4.6 Struktur 15](#_Toc138774268)

[4.7 Name und Logo 16](#_Toc138774269)

[4.8 3D Modelle 17](#_Toc138774270)

[5 Entscheiden 18](#_Toc138774271)

[5.1 3D Modelle 18](#_Toc138774272)

[5.2 Deployment Github Pages 18](#_Toc138774273)

[5.2.1 Die Vorteile der Nutzung von Github Pages sind: 18](#_Toc138774274)

[5.3 Farben 19](#_Toc138774275)

[5.4 Logo 19](#_Toc138774276)

[5.5 Programmiersprache React 19](#_Toc138774277)

[5.6 Libaries und Packages 20](#_Toc138774278)

[5.6.1 React Three Fiber 20](#_Toc138774279)

[5.6.2 Tailwind CSS 20](#_Toc138774280)

[5.6.3 Framer Motion 20](#_Toc138774281)

[6 Realisieren 22](#_Toc138774282)

[6.1 Anwendung initialisieren 22](#_Toc138774283)

[6.2 Webseite Komponente 23](#_Toc138774284)

[6.2.1 Navigationsleiste 23](#_Toc138774285)

[6.2.2 Hero 23](#_Toc138774286)

[6.2.3 Skills 25](#_Toc138774287)

[6.2.4 Work 25](#_Toc138774288)

[6.2.5 Contact 26](#_Toc138774289)

[6.2.6 SectionWrapper 27](#_Toc138774290)

[6.2.7 CanvasLoader 28](#_Toc138774291)

[6.2.8 Stern-Background 28](#_Toc138774292)

[6.3 Daten und Content 29](#_Toc138774293)

[6.4 Deployment 29](#_Toc138774294)

[6.4.1 Probleme 29](#_Toc138774295)

[6.4.2 Lösung 30](#_Toc138774296)

[7 Kontrollieren 31](#_Toc138774297)

[7.1 Testfall Figuren bewegen 31](#_Toc138774298)

[7.2 Testfall Links öffnen 31](#_Toc138774299)

[7.3 Testfälle Nachricht senden 31](#_Toc138774300)

[7.4 Testfall Navigation auf der Webseite 32](#_Toc138774301)

[8 Auswerten 33](#_Toc138774302)

[9 Quellenverzeichnis 34](#_Toc138774303)

[10 Abbildungsverzeichnis 35](#_Toc138774304)

# Änderungstabelle

|  |  |  |  |
| --- | --- | --- | --- |
| Datum | Wer? | Was? | Geprüft? |

|  |  |  |  |
| --- | --- | --- | --- |
| 05.06.2023 | Jann Fanzun | Dokument erstellt und Grundstein. |  |
| 05.06.2023 | Jann Fanzun | Überschriften gesetzt. |  |
| 06.06.2023 | Jann Fanzun | Planen, Informieren bearbeitet |  |
| 07.06.2023 | Jann Fanzun | Flow Chart, Ordnerstruktur |  |
| 07.06.2023 | Jann Fanzun | Wireframe, Entscheiden |  |
| 12.06.2023 | Jann Fanzun | Realisieren |  |
| 20.06.2003 | Jann Fanzun | Planen fertig |  |
| 21.06.2023 | Jann Fanzun | Entscheiden begonnen |  |
| 26.06.2023 | Jann Fanzun | Entscheiden abgeschlossen |  |
| 26.06.2023 | Jann Fanzun | Realisieren begonnen |  |
| 26.06.2023 | Jann Fanzun | Kontrollieren begonnen |  |
| 27.06.2023 | Jann Fanzun | Kontrollieren abgeschlossen |  |
| 27.06.2023 | Jann Fanzun | Realisieren abgeschlossen |  |

# Projektidee

## Projektauftrag

Das Ziel dieses Projekts ist die Entwicklung einer modernen, interaktiven und ansprechenden 3D-Entwicklerwebsite. Diese Plattform bietet Kunden die Möglichkeit, den Entwickler kennenzulernen und beeindruckenden Projekte zu verfolgen. Die Website wird nicht nur auf Desktop-Computern, sondern auch auf mobilen Geräten optimal funktionieren, da sie responsive gestaltet wird.

## Ausgangslage

Ich plane, eine persönliche Webseite zu erstellen, um meine Fähigkeiten und bisherigen Projekte zu präsentieren. Durch diese Webseite möchte ich nicht nur meine Arbeit zeigen, sondern auch eine neue Programmiersprache erlernen und meine Fortschritte dokumentieren. Mein Ziel ist es, eine moderne und ansprechende Webseite zu entwickeln, die meine Fähigkeiten als Entwickler optimal präsentiert. Um potenziellen Interessenten die Möglichkeit zu geben, mich zu kontaktieren, werde ich auch ein Kontaktformular auf der Webseite bereitstellen.

## Ziele

|  |  |
| --- | --- |
| Projektziele (Erwartete Resultate. Forderungen und Wünsche) | Prior.1 |
| Persönliche Webseite zu erstellen, auf der ich meine Fähigkeiten und bisherigen Projekte präsentieren kann. | A |
| Eine neue Programmiersprache zu erlernen. Durch die Entwicklung dieser Webseite möchte ich meine Kenntnisse in dieser Sprache erweitern und praktische Erfahrungen sammeln. | A |
| Die Webseite soll auch dazu dienen, meine Fortschritte und Können als Entwickler festzuhalten. Ich plane, regelmässig über meine Lernerfolge, Herausforderungen und neuen Fähigkeiten zu berichten. | A |
| Eine Webseite zu erstellen, die modern und ästhetisch ansprechend ist. Sie sollte benutzerfreundlich gestaltet sein und ein positives Nutzererlebnis bieten. | B |
| Kundenmeinungen / Testimonials | C |
| **Vorgehensziele** (Forderungen und Rahmenbedingungen) | **Prior.1** |
| Es ist wichtig, dass das Kontaktformular einwandfrei funktioniert und potenziellen Interessenten eine einfache Möglichkeit bietet, mich zu kontaktieren. | B |
| Die Webseite sollte auf verschiedenen Geräten und Bildschirmgrössen gut funktionieren. Ein responsives Design ist erforderlich. | B |

|  |  |
| --- | --- |
| Lernziele (Welche Kompetenzen werden angestrebt?) | Prior.1 |
| Beherrschung einer neuen Programmiersprache 🡪 Three.js (React), Tailwind | A |
| Webdesign | A |
| Projektmanagement | A |
| 3D Grafiken und Animationen erlernen 🡪 Framer Motion | B |
| Kontaktformular 🡪 E-Mail JS | C |

|  |  |  |  |
| --- | --- | --- | --- |
| 1) A = Must have | B = Starkes Wunschziel, wenn irgendwie möglich | C = “Nice to have“ |  |

# Informieren

## Planungstechnologien

Es gibt viele verschiedene Tools um zu Planen. Einige der beliebtesten Optionen sind:

* **Canva**: Canva ist ein webbasiertes Design-Tool, das eine Vielzahl von Vorlagen und Funktionen bietet. Es ermöglicht das Erstellen von ansprechenden Grafiken, Infografiken und Layouts, die bei der visuellen Planung und Gestaltung eines Projekts hilfreich sein können.
* **ClickUp**: ClickUp ist eine Projektmanagement-Plattform, die verschiedene Tools und Funktionen zur Planung, Organisation und Zusammenarbeit bietet. Es ermöglicht das Erstellen von Aufgaben, das Zuweisen von Verantwortlichkeiten, das Verfolgen des Fortschritts und die Kommunikation innerhalb des Teams.
* **Word**: Microsoft Word ist eine weit verbreitete Textverarbeitungssoftware, die oft für die Erstellung von Projektdokumentationen, Berichten und anderen schriftlichen Materialien verwendet wird. Es bietet Funktionen zur Formatierung, Bearbeitung und Freigabe von Dokumenten.
* **Notion**: Notion ist eine All-in-One-Plattform für Projektmanagement, Notizen und Teamkollaboration. Es ermöglicht das Erstellen von individuellen Dashboards, Aufgabenlisten, Notizen und Dokumentationen. Notion bietet flexible Möglichkeiten zur Organisation und Planung von Projekten.

Diese Planungstechnologien bieten verschiedene Funktionen und Vorteile, um die Planung, Organisation und Zusammenarbeit in einem Projekt effektiv zu unterstützen

## Entwicklungsumgebung

Es gibt viele verschiedene IDE's, die für die Entwicklung von Software in verschiedenen Programmiersprachen verwendet werden können. Einige der beliebtesten Optionen sind:

* **Visual Studio Code**: Eine IDE von Microsoft, die für die Entwicklung von Anwendungen in verschiedenen Sprachen verwendet werden kann, darunter C#, C++, Visual Basic, React und F#.
* **Eclipse**: Eine freie und quelloffene IDE, die für die Entwicklung von Anwendungen in verschiedenen Sprachen verwendet werden kann, darunter Java, C++ und PHP.
* **IntelliJ IDEA**: Eine kommerzielle IDE von JetBrains, die für die Entwicklung von Anwendungen in verschiedenen Sprachen verwendet werden kann, darunter Java, Kotlin, Python, React und Ruby.
* **Xcode**: Eine IDE von Apple, die für die Entwicklung von Anwendungen für das iOS- und macOS-Betriebssystem verwendet wird.

## Frontend

Es gibt viele verschiedene Technologien und Frameworks, die für die Entwicklung von Frontend-Anwendungen verwendet werden können. Einige der beliebtesten Optionen sind:

* **HTML**: Eine Markup-Sprache, die verwendet wird, um die Struktur und den Inhalt von Webseiten zu definieren.
* **React**: Ein JavaScript-Framework, das von Facebook entwickelt wurde und häufig für die Entwicklung von Webanwendungen verwendet wird.
* **Angular**: Ein JavaScript-Framework, das von Google entwickelt wurde und häufig für die Entwicklung von Webanwendungen verwendet wird.
* **Vue.js**: Ein JavaScript-Framework, das für die Entwicklung von Benutzeroberflächen verwendet wird.

Die Schwierigkeit darin besteht darin, beim Aneignen der Sprachen.

## Deployment Methoden

Es gibt viele verschiedene Technologien und Anbieter, für das Deployment und hosten von Webseiten. Einige der beliebtesten Optionen sind:

* **Netlify** (https://www.netlify.com/): Netlify bietet ein einfaches und leistungsstarkes Hosting für statische Webseiten. Es ermöglicht dir, deine Webseite direkt aus einem Git-Repository zu deployen und bietet Funktionen wie Continuous Deployment, automatische Skalierung und ein globales CDN.
* **Vercel** (<https://www.vercel.com/>): Vercel ist eine Plattform-as-a-Service (PaaS), die dir ermöglicht, Webanwendungen schnell und einfach zu deployen. Es unterstützt verschiedene Programmiersprachen und Frameworks und bietet Funktionen wie Skalierung, Logging und Überwachung.
* **GitHub Pages** (https://pages.github.com/): GitHub Pages ist eine integrierte Funktion von GitHub, mit der du statische Webseiten direkt aus einem GitHub-Repository hosten kannst. Es ist einfach einzurichten und bietet eine nahtlose Integration mit Git. Du kannst eine eigene Domain verwenden oder eine Subdomain von github.io nutzen.
* **Hostinger** (https://hostinger.com/): Hostinger ist ein bekannter Webhosting-Anbieter, der eine breite Palette von Hosting-Optionen bietet. Sie bieten eine einfache und benutzerfreundliche Plattform zum Deployment von Webseiten.

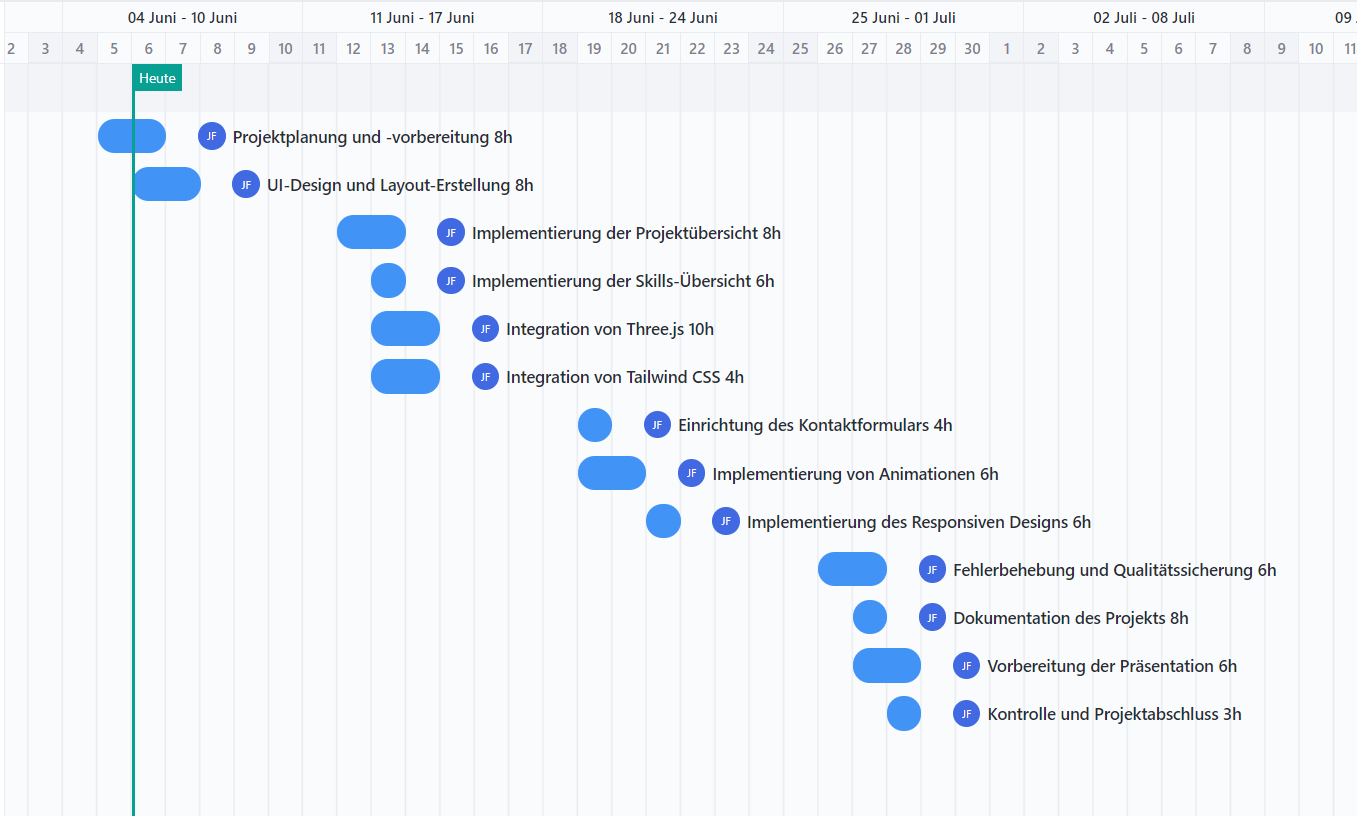
Der Unterschied besteht darin vom Preis der Domain, der Möglichkeit, umsonst zu deployen und der Flexibilität mit Github Repositories.

# Planen

## Arbeitspakete

|  |  |  |
| --- | --- | --- |
| Was | Beschreibung | Zeit |
| Projektplanung und -vorbereitung | Erstellung des Zeitplans, Ressourcenplanung, Risikoanalyse | **8h** |
| UI-Design und Layout-Erstellung | Gestaltung des UIs, Erstellung von Wireframes und Mockups | **8h** |
| Implementierung der Projektübersicht | Umsetzung der Projektübersichtsseite mit den entsprechenden Informationen und Funktionen | **8h** |
| Implementierung der Skills-Übersicht | Umsetzung der Skills-Übersichtsseite zur Darstellung der Fähigkeiten und Erfahrungen | **6h** |
| Integration von Three.js | Einbindung von Three.js für die 3D-Darstellung der Portfolio-Modelle | **10h** |
| Integration von Tailwind CSS | Einbindung von Tailwind CSS für das Styling und die Anpassung des Designs | **4h** |
| Einrichtung des Kontaktformulars | Konfiguration des Kontaktformulars mit EmailJS zur Kommunikation mit dem Website-Inhaber | **4h** |
| Implementierung von Animationen | Einbau von Framer Motion für ansprechende Animationen auf der Website | **6h** |
| Implementierung des Responsiven Designs | Anpassung der Website für verschiedene Bildschirmgrössen und Geräte | **6h** |
| Fehlerbehebung und Qualitätssicherung | Testen der Funktionalitäten, Beheben von Fehlern und Sicherstellung der Qualität | **6h** |
| Dokumentation des Projekts | Erstellung einer detaillierten Dokumentation über den Code, die Funktionalitäten und die Implementierung | **8h** |
| Vorbereitung der Präsentation | Vorbereitung der Präsentation des Projekts, Erstellung von Präsentationsmaterialien | **6h** |
| Kontrolle und Projektabschluss | Abschlussarbeiten, Überprüfung des Projekts, finale Anpassungen | **3h** |
| Gesamt | - | **83h** |

## Gantt-Diagramm



1 Gantt Diagramm

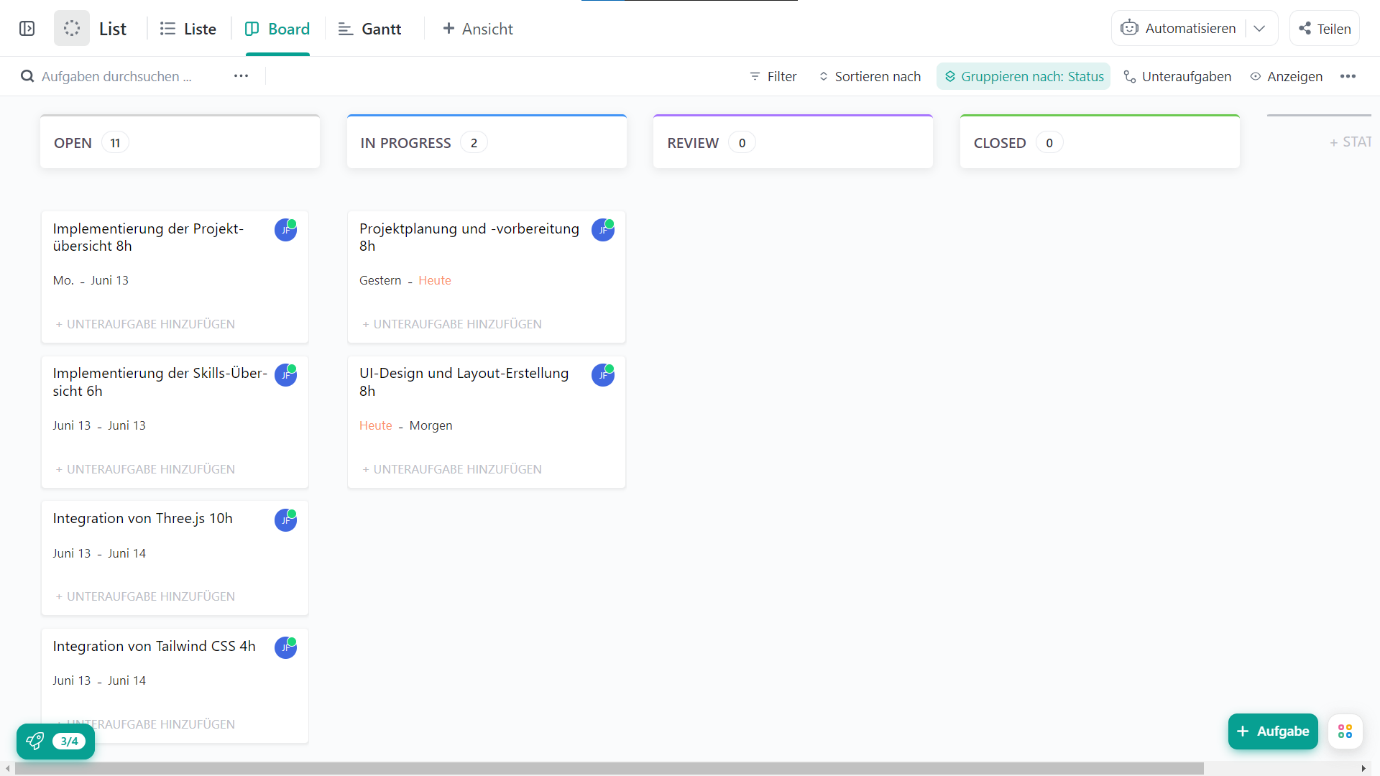
Das Gantt-Diagramm dient als nützliches Werkzeug, um den Überblick über das Projekt zu behalten und sich daran zu orientieren. Es ermöglicht mir, den Fortschritt der einzelnen Aufgaben im Zeitverlauf zu verfolgen und sicherzustellen, dass das Projekt im vorgegebenen Zeitrahmen bleibt. Durch die visuelle Darstellung der Aufgaben und deren zeitlicher Zuordnung hilft das Gantt-Diagramm dabei, Prioritäten zu setzen und Engpässe frühzeitig zu erkennen.

Die Gesamtzeit von 83 Stunden im Gantt-Diagramm enthält auch ca. 10 Stunden Pufferzeit, die für unvorhergesehene Ereignisse oder zusätzliche Anpassungen reserviert sind. Dieser Puffer bietet Flexibilität, um auf unerwartete Hindernisse oder Änderungen reagieren zu können, ohne den gesamten Zeitplan zu gefährden.

## Kanban Board

Das Kanban Board ist ein präzises und hilfreiches Instrument, um den Fortschritt des Projekts im Detail zu verfolgen. Im Gegensatz zum Gantt-Diagramm ermöglicht das Kanban Board eine detaillierte Aufgabenverwaltung, indem es die einzelnen Aufgaben in Karten oder Listen organisiert.

Mit dem Kanban Board können Aufgaben klar priorisiert, vorangetrieben und abgeschlossen werden. Es bietet eine visuelle Darstellung des Prozesses, indem Aufgaben von einer Spalte zur nächsten verschoben werden, um ihren Fortschritt anzuzeigen. Dadurch wird es einfach, den aktuellen Status der Aufgaben zu verfolgen und Engpässe oder Verzögerungen frühzeitig zu erkennen.



2 Kanban Board

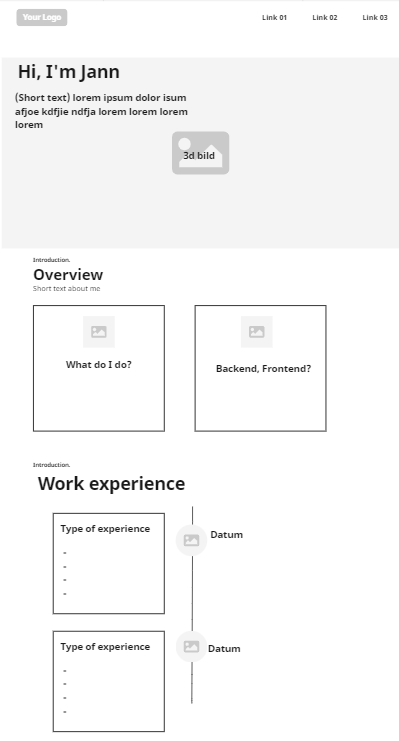
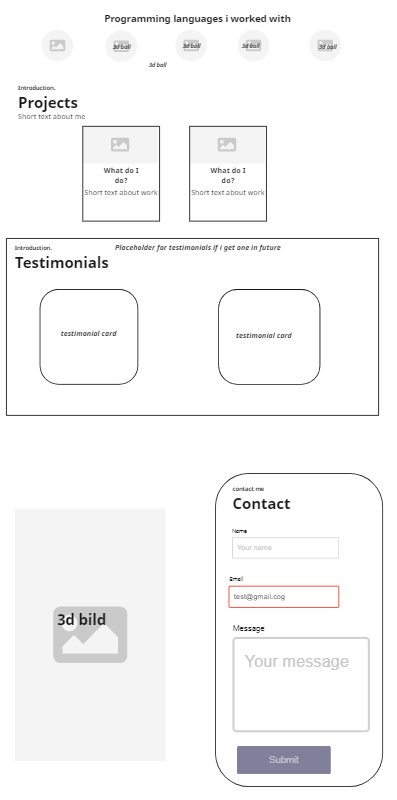
## Wireframe

Die Planung meines Wireframes bietet mehrere Vorteile:

1. **Klarheit**: Ein Wireframe definiert den Aufbau und die Struktur der Benutzeroberfläche klar und verständlich.
2. **Benutzerfreundlichkeit**: Durch die Berücksichtigung von Benutzerinteraktionen verbessert der Wireframe die Benutzerfreundlichkeit.
3. **Effizienz**: Frühzeitige Diskussionen über Design und Funktionalität sparen Zeit und Ressourcen.
4. **Kommunikation**: Der Wireframe dient als Kommunikationsmittel zwischen Stakeholdern, Designern und Entwicklern.
5. **Flexibilität**: Der Wireframe ermöglicht Änderungen und Anpassungen vor der eigentlichen Entwicklung.

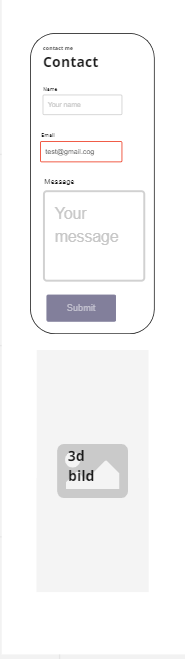
Die Planung meines Wireframes legt die Grundlage für ein erfolgreiches Design und eine reibungslose Entwicklung, um meine Ziele zu erreichen und die Benutzererfahrung zu optimieren. Dabei habe ich ein Wireframe für die Laptop Ansicht und ein Wireframe für die responsive Ansicht auf dem Telefon designet.

### Laptop 16:9



3 Wireframe Laptop

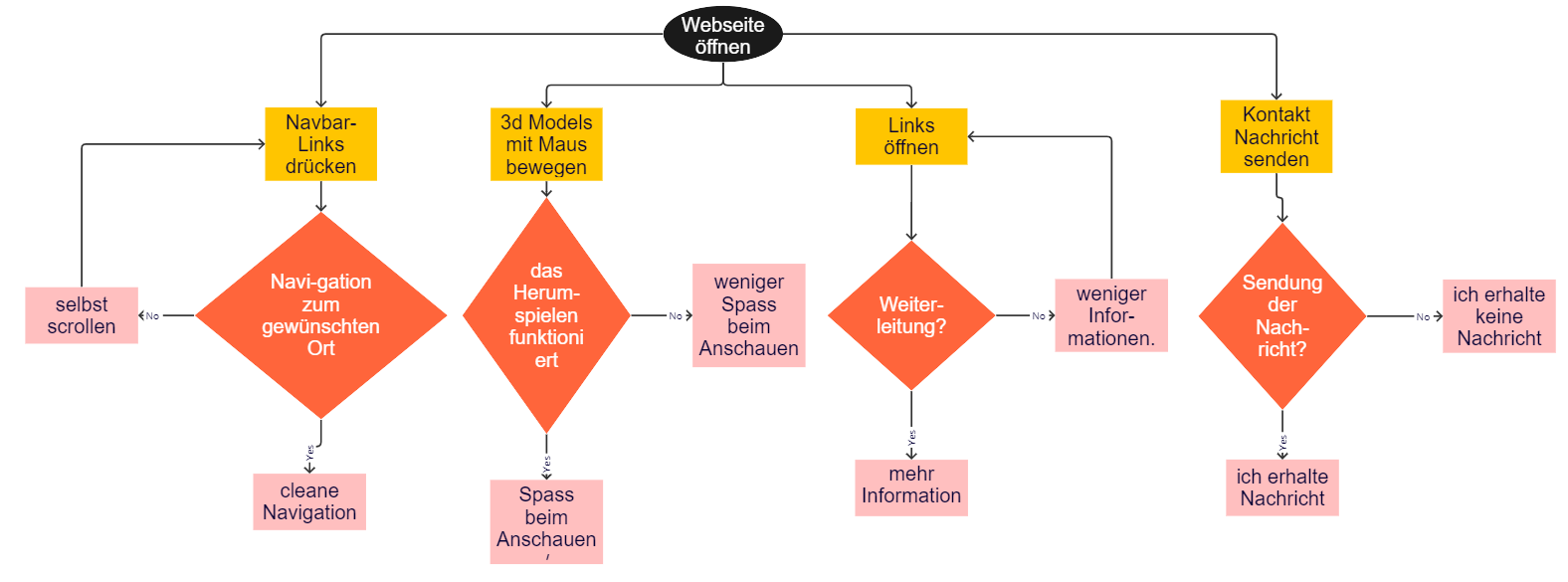
### Telefon 19.5:9



4 Wireframe Telefon

## Flow Chart

5 Flow Chart



1. **Start**: Der Benutzer öffnet die Webseite und landet auf der Startseite.
2. **Navigation**: Der Benutzer kann die Navbar verwenden, um durch die Webseite zu navigieren. Die Navbar enthält verschiedene Links zu den verschiedenen Bereichen der Webseite, z. B. Über Uns, Galerie, Kontakt usw.
3. **3D-Modelle**: Innerhalb der Webseite gibt es 3D-Modelle, die der Benutzer bewegen und interaktiv erkunden kann. Diese Modelle können mithilfe von Mausbewegungen oder Touch-Gesten gesteuert werden.
4. **Links**: Auf der Webseite gibt es auch Links zu weiteren Inhalten, z. B. zu relevanten Artikeln oder verwandten Ressourcen. Der Benutzer kann auf diese Links klicken, um weitere Informationen zu erhalten.
5. **Kontaktformular**: Die Webseite enthält ein Kontaktformular, das es dem Benutzer ermöglicht, eine Nachricht zu senden. Der Benutzer kann das Kontaktformular ausfüllen und absenden, um mit dem Website-Betreiber in Kontakt zu treten.
6. **Nachricht senden**: Nachdem der Benutzer das Kontaktformular ausgefüllt und abgesendet hat, wird die Nachricht an den Empfänger geschickt. Der Empfänger erhält die Nachricht und kann darauf antworten.
7. **Ende**: Der Flowchart endet hier, und der Benutzer kann weiterhin die Webseite erkunden, die 3D-Modelle bewegen, Links öffnen und bei Bedarf erneut das Kontaktformular verwenden.

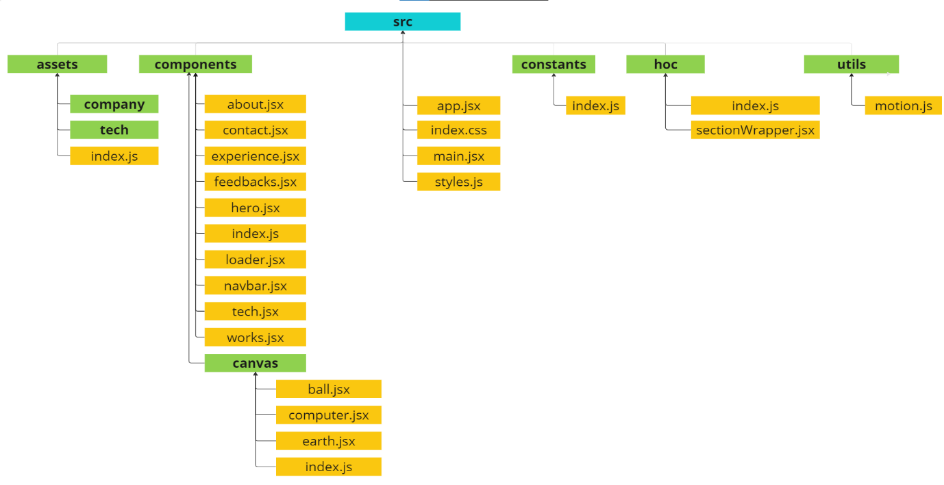
## Struktur

Die Ordnerstruktur besteht aus einem übergeordneten Ordner, der die Farbe Blau hat. Innerhalb dieses blauen Ordners befindet sich ein grüner Ordner. Die Farbe Blau repräsentiert den obersten Ordner, während die grüne Farbe für den Unterordner steht.

Im blauen Oberordner "src" sind insgesamt 8 Unterordner und 23 Dateien enthalten. Es handelt sich um eine strukturierte Sammlung von Informationen und Ressourcen. Im components Ordner sind alle «jsx» Files, in denen ich programmieren werde.

Die Index.js Files sind dafür da, dass ich meine Bilder und Texte importieren und anschliessend exportieren kann. Somit exportiere ich alles in den Ordner Constants und so habe ich alle meine Daten und Texte in einem File gespeichert.

Des Weiteren gibt es zwei Unterordner mit den Namen "company" und "tech". In diesen Unterordnern sind Bilder oder Grafiken enthalten, die im Zusammenhang mit dem jeweiligen Thema stehen.



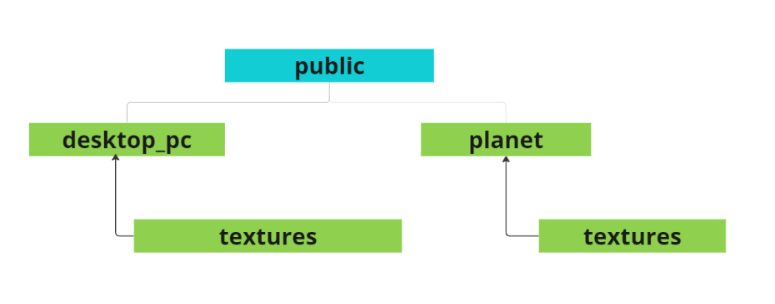
6 Struktur src

Die Ordnerstruktur besteht aus einem blauen Oberordner namens "public", der als Wurzelverzeichnis dient. Innerhalb dieses blauen Ordners befinden sich zwei Unterordner und in diesen Unterordner gibt es je einen weiteren Unterordner namens "textures".

Diese beiden Unterordner dienen zur Organisation von Bilddateien für 3D-Modelle.

Es ist zu beachten, dass die Farbe Blau den Oberordner kennzeichnet und die Farbe Grün den Unterordner.

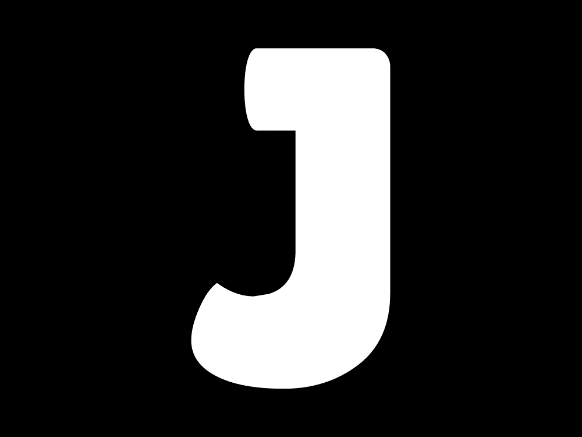
7 Struktur public



## Name und Logo

Ich habe mich dazu entschieden, da ich eine Portfolio Website mache, dass mein Websitename Jann Fanzun sein wird. Ich werde meine Website deployen und somit eine Domain kaufen.

Zur Website brauche ich ein passendes Logo beziehungsweise ein passendes Favicon. Da ich in der Website 3D-Modelle einbaue, brauche ich nur ein kleines Icon in der Navigationsleiste, Fusszeile und im Browser.



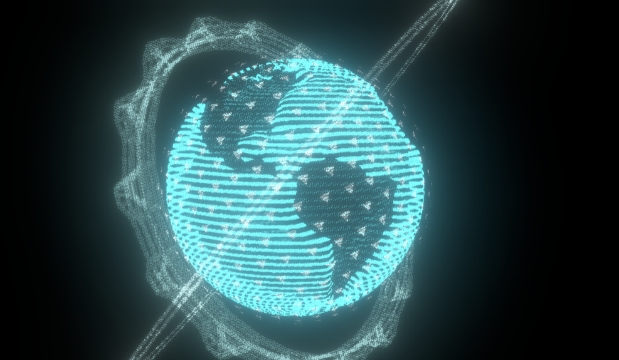
8 potenzielle Logos und Favicons

## 3D Modelle

Auf einer Informatiker-Portfolio-Website können 3D-Modelle eine grossartige User Experience bieten. Diese Modelle können sich bewegen und drehen, um den Besuchern eine interaktive Betrachtung zu ermöglichen. Ich plane, ein 3D Modell auf der Landingpage zu haben und eines beim Kontaktformular.



9 PC 3d Modelle



10 Weltkugel 3d Modelle

# Entscheiden

## 3D Modelle

Ich habe mich für die «feinen» Modelle entschieden. Ich möchte moderne, aber nicht zu übertriebene Modelle auf meiner Webseite haben. Das PC-Setup wird auf der Landingpage ersichtlich sein und die Weltkugel beim Kontaktformular.



11 definitive 3d Figuren

## Deployment Github Pages

Ich habe mich für das Deployment meiner Webseite für Github Pages entschieden, da es mehrere Vorteile bietet:

* Einfache Einrichtung und Bereitstellung der Webseite
* Kostenlose Hosting-Lösung
* Nahtlose Integration mit vorhandenen GitHub Repositories

Da ich bereits eine Webseite erfolgreich mit Github Pages deployt habe und mein Projekt als GitHub Repository vorliegt, ist es die am wenigsten aufwendige Option, um meine Dateien zu deployen.

### Die Vorteile der Nutzung von Github Pages sind:

* Einfache Einrichtung und Konfiguration
* Automatische Aktualisierung der Webseite bei Änderungen im Repository
* Möglichkeit zur Verwendung von benutzerdefinierten Domains

Durch die Wahl von Github Pages für das Deployment meiner Webseite kann ich eine kostengünstige und effiziente Lösung nutzen, um meine Inhalte online verfügbar zu machn.

## Farben

In Bezug auf meine Webseite habe ich mich dafür entschieden, zwei schlichte Farben sowie eine auffällige, stechende Farbe zu verwenden. Die beiden schlichten Farben, die ich für mein Webdesign ausgewählt habe, sind Schwarz und Weiss. Diese klassische Farbkombination strahlt Eleganz, Klarheit und Zeitlosigkeit aus, was meinem Webprojekt ein professionelles und ansprechendes Erscheinungsbild verleihen wird.

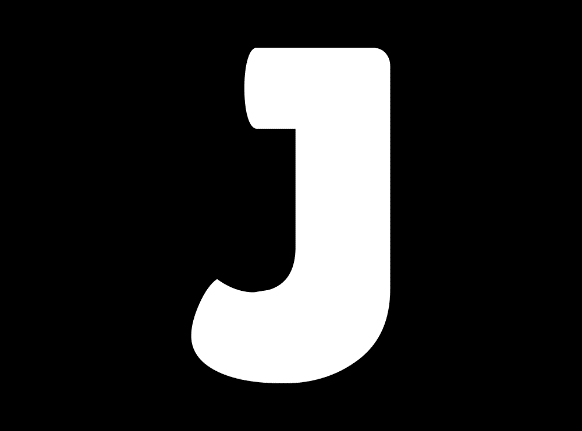
schwarz

weiss

violett

## Logo

Ich habe mich für das weisse «J» und das schwarze «J» entschieden. In meiner Vorstellung möchte ich das weisse «J» in dem Browser als Favicon haben und das schwarze «J» soll in meiner Navigationsleiste und in der Fusszeile sein.



12 definitive Logos

## Programmiersprache React

Nach sorgfältiger Überlegung habe ich mich für React als Programmiersprache entschieden. Hier sind die Gründe dafür:

* Solide Kenntnisse in JavaScript und HTML werden genutzt.
* Aufteilung der Anwendung in wiederverwendbare Komponenten für organisierten und skalierbaren Code.
* Leistungsstarke und effiziente Benutzeroberflächen dank Virtual DOM.
* Aktive Entwicklergemeinschaft bietet Unterstützung und Ressourcen.
* Flexibles Importieren von Komponenten und Code für Wiederverwendbarkeit und effiziente Entwicklung.

Durch die Entscheidung, mit React zu arbeiten, kann ich meine bestehenden Kenntnisse nutzen und von den Vorteilen einer flexiblen Entwicklung und einer breiten Bibliothek profitieren.

## Libaries und Packages

### React Three Fiber

Für die Umsetzung von Anwendungen wird die Verwendung von React Three Fiber empfohlen. React Three Fiber ist eine Library, die in Verbindung mit React einfach Animationen und 3D-Modelle einbinden kann. Hier sind Vorteile von React Three Fiber:

* **Einfache Einbindung**: Durch die Verwendung von React-Komponenten wird die Erstellung und Steuerung der 3D-Szene vereinfacht und besser verständlich.
* **Gute Leistung**: React Three Fiber basiert auf Three.js, einer Bibliothek für 3D-Rendering mit WebGL. Dadurch werden 3D-Inhalte schnell und flüssig dargestellt.
* **Weniger Code**: Mit React Three Fiber kann man komplexe 3D-Interaktionen und Animationen mit weniger Code realisieren. Dies spart Zeit und Aufwand.

Zusammenfassend ermöglicht React Three Fiber die einfache Integration von 3D-Visualisierungen in React-Anwendungen.

### Tailwind CSS

Entscheidung für Tailwind CSS in meiner Applikation:

* Einfache Verwendung durch direkte Klassennamen statt separate CSS-Dateien.
* Weniger Dateierstellung, da vorgefertigte Klassen genutzt werden können.
* Effiziente Entwicklung durch schnelle Anwendung und Anpassung der Stile.
* Flexibilität und Anpassungsmöglichkeiten durch eigene Klassen und Kombinationen.

Tailwind CSS ist die optimale Wahl für meine Applikation aufgrund seiner einfachen Verwendung, reduzierten Dateierstellung, effizienten Entwicklung, Flexibilität und konsistenten Gestaltung.

### Framer Motion

Entscheidung für die Verwendung von Framer Motion:

* **Einfache Handhabung**: Framer Motion ist äusserst benutzerfreundlich und einfach zu verwenden.
* **Weniger Dateien**: Im Vergleich zu anderen Bibliotheken erfordert Framer Motion weniger Dateien und Konfigurationen, was die Entwicklungszeit verkürzt.
* **Animierte Übergänge**: Framer Motion bietet eine Vielzahl von Animations- und Übergangseffekten, die das Erscheinungsbild der Anwendung verbessern.
* **Gute Dokumentation**: Framer Motion verfügt über eine umfangreiche und gut strukturierte Dokumentation, die Entwicklern hilft, schnell und effizient damit zu arbeiten.
* **Flexibilität**: Die Bibliothek bietet eine hohe Flexibilität bei der Erstellung von Animationen und ermöglicht es, individuelle Gestaltungskonzepte umzusetzen.

Die Entscheidung, Framer Motion zur Realisierung der Anwendung zu verwenden, basiert auf der einfachen Handhabung, der geringen Anzahl von Dateien, den animierten Übergängen, der guten Dokumentation, der Flexibilität und der Unterstützung für verschiedene Plattformen.

# Realisieren

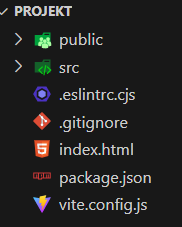
## Anwendung initialisieren

Folgende Packages habe ich für meine Anwendung installiert:

* `***npm create vite@latest ./ -- --template react***`: Erstellt ein neues React-Projekt mit Vite als Build-Tool.
* `***npm install -D tailwindcss***`: Installiert Tailwind CSS als Entwicklungsabhängigkeit.
* ***`npx tailwindcss init***`: Initialisiert die Tailwind CSS-Konfigurationsdatei.
* `***npm install @react-three/fiber***`: Ermöglicht die Verwendung von WebGL-basierten 3D-Grafiken in React-Anwendungen.
* `***npm install @react-three/drei***`: Eine Sammlung von wiederverwendbaren 3D-Grafikkomponenten für React und Three.js.
* `***npm install maath***`: Ein mathematisches Utility-Package für mögliche Berechnungen.
* `***npm install react-tilt***`: Bietet eine React-Komponente für parallaxe Neigungseffekte.
* `***npm install react-vertical-timeline-component***`: Erstellt vertikale Zeitachsenkomponenten in React.
* `***npm install @emailjs/browser***`: Ermöglicht das Versenden von E-Mails über JavaScript und den E-MailJS-Dienst.
* `***npm install framer-motion***`: Bietet eine API für animierte Übergänge und Animationen in React-Anwendungen.
* `***npm install react-router-dom***`: Bereitstellung von Routing-Funktionen für die Navigation zwischen Seiten in React-Anwendungen.

Für die Konfiguration von Tailwind CSS wurden zusätzlich folgende Packages installiert:

* `***npm install -D postcss autoprefixer***`: Entwicklungsabhängigkeiten für die Verarbeitung von CSS-Dateien mit PostCSS und Autoprefixer.

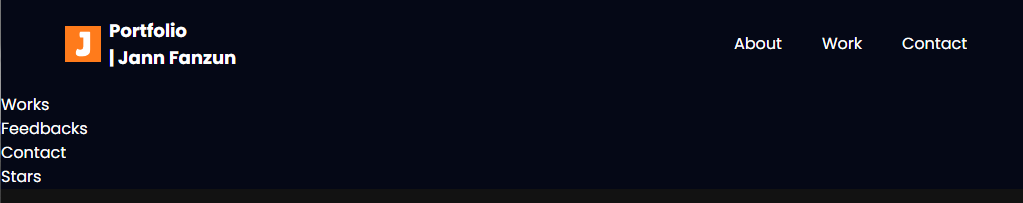


13 Ordner & Files nach Initialisierung

* `***npx tailwindcss init -p***`: Konfiguriert Tailwind CSS mit PostCSS und Autoprefixer für korrekte Funktionalität.

## Webseite Komponente

### Navigationsleiste



14 Navigationsleiste



15 Entwicklung der Navigationsleiste

Beim Programmieren der Navigationsleiste, habe ich auf der linken Seite mein Logo und mein Name geschrieben und auf der rechten Seite meine Links der Sections aufgereiht.

Zu Beginn war das Bild zu gross und der Text neben dem Bild wurde nicht richtig skaliert. Mithilfe folgendem Codestück konnte ich es oben links skalieren. Siehe Bild 14 und 15.

<img src={logoblack} alt='logo' className='w-9 h-9 object-contain rounded-full' />

          <p className='text-white text-[18px] font-bold cursor-pointer flex '>

            Jann Fanzun &nbsp;

            <span className='ld:block hidden'> | Portfolio</span>

          </p>

Beim nächsten Codestück sieht man, wie ich meine Links von den Constants hole. Ich habe zuvor die ***navLinks*** importiert und nun kann ich mit wenig Code, den ***link.title***und die ***link.id*** holen, welche die Navigation steuert durch den Link Tag. Siehe Bild 15.

<ul className='list-none hidden md:flex flex-row gap-10 cursor-pointer'>

          {navLinks.map((link) => (

            <li

            key={link.id}

            className={`${

              active === link.title ? "text-white" : "text-secondary"

            } hover:text-white text-[18px] font-medium cursor-pointer`}

            onClick={() => setActive(link.title)}

            >

              <a href={`#${link.id}`}>{link.title}</a>

            </li>

          ))}

        </ul>

### Hero

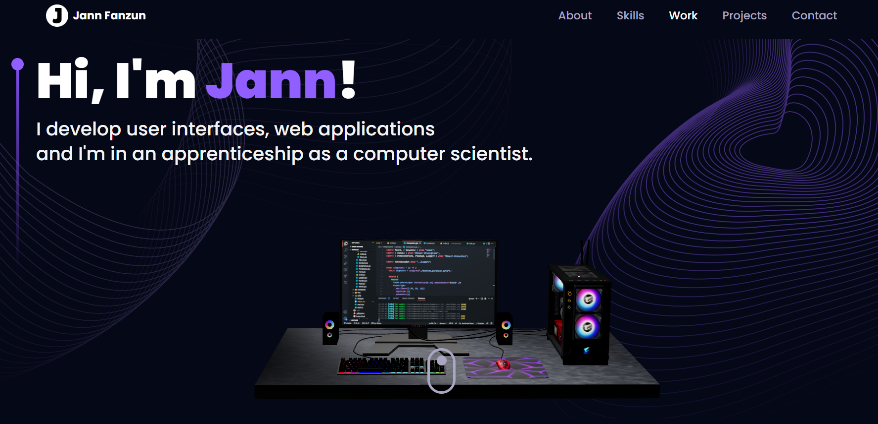
In der Hero Section meiner Webseite habe ich erfolgreich mein erstes 3D-Modell eingebaut. Um das GLTF-File einzufügen, musste ich die folgenden Importe verwenden:

import { OrbitControls, Preload, useGLTF } from "@react-three/drei";

Um das GLTF-Modell in meinem Code einzufügen, musste ich es in einen ***<mesh></mesh>***-Tag platzieren. Dabei gab es einige Aspekte, die ich beachten musste:

* Die Verwendung von ***hemisphereLight*** für eine gleichmässige Beleuchtung.
* Die Verwendung von ***spotLight*** für eine gezielte Lichtquelle.
* Die Verwendung von ***pointLight*** für punktuelle Beleuchtung.
* Die ***Positionierung*** des Modells innerhalb des Szenengraphen.
* Die Verwendung von ***OrbitControls*** zur Interaktion mit dem 3D-Modell.
* Die Verwendung des ***Canvas***-Elements als Container für das 3D-Rendering.

Durch das Verständnis und Experimentieren mit diesen Konzepten war es mir möglich, mein 3D-Modell erfolgreich zu platzieren und sichtbar zu machen.

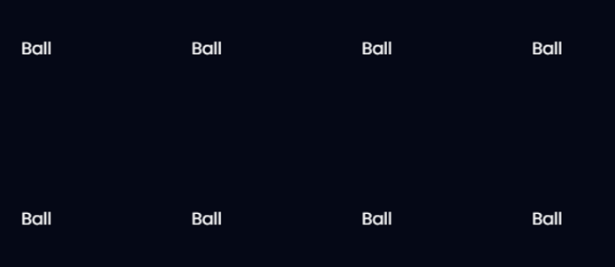


16 3D Modell

### Skills

Der Abschnitt «Skills» schafft eine Szene mit einem schwebenden Objekt, Lichtern, Materialien und einem zusätzlichen Designelement.

* Das Objekt selbst hat eine bestimmte Form, ähnlich einem geometrischen Objekt namens "**Icosahedron**".
* Es gibt zwei Arten von Lichtern in der Szene:
  + Ein schwaches **Umgebungslicht**, das die Szene insgesamt erhellt.
  + Eine gerichtete **Lichtquelle**, die von einer bestimmten Position aus auf das Objekt scheint.
* Auf dem Objekt befindet sich ein zusätzliches **Designelement**, das als "***Decal***" bezeichnet wird.
  + Das Design hat eine bestimmte Position, Rotation und Skalierung.
  + Es verwendet eine spezielle Textur, um das Aussehen des Designs zu erzeugen.



17 Entwicklung von den Skills-Bällen



18 Fertigstellung der Skills-Bälle

### Work

Mein Vorgehen bei der Section «Work» war sehr strukturiert und durchdacht, da das der anspruchsvollste Teil meines Projektes war:

* Zuerst werden die benötigten Module und Bibliotheken importiert.
* Dann wird eine Komponente namens "***ExperienceCard***" definiert, die eine einzelne Erfahrung repräsentiert.
  + Diese Komponente zeigt die Informationen der Erfahrung in einer Kartenansicht an.
  + Dazu wird die "***VerticalTimelineElement***"-Komponente verwendet.
  + Die Karte enthält den Titel der Erfahrung, den Namen des Unternehmens und eine Liste von Punkten.
* Anschliessend wird die Hauptkomponente "***Experience***" definiert.
  + Diese Komponente repräsentiert die gesamte Erfahrungssektion.
  + In der Sektion wird eine vertikale Zeitleiste angezeigt.
  + Für jede Erfahrung wird eine "***ExperienceCard***"-Komponente erstellt und in der Zeitleiste platziert.

Im Progress hat mir das ***VerticalTimelineElement*** am meisten Arbeit gekostet. Mein Code erstellt ein Element für eine vertikale Zeitleiste. Er hat verschiedene Eigenschaften und Stile, die definiert sind.

* Das ***contentStyle*** definiert den Hintergrund und die Textfarbe des Inhaltsbereichs des Elements.
* ***contentArrowStyle*** definiert den Stil des Pfeils, der den Inhaltspunkt mit dem Zeitpunkt verbindet.
* Das ***date-Attribut*** stellt das Datum dar, das mit diesem Element verbunden ist.
* Das ***iconStyle*** definiert den Hintergrundstil des Icons, das in diesem Element angezeigt wird.



19 Work Abschnitt

* Das ***icon-Attribut*** enthält den eigentlichen Inhalt des Icons, das in Form eines Bildes angezeigt wird.

Zusammengefasst erstellt mein Code ein Element für eine vertikale Zeitleiste, das mit einem Datum und einem Icon versehen ist.

<VerticalTimelineElement

      contentStyle={{

        background: "#1d1836",

        color: "#fff",

      }}

      contentArrowStyle={{ borderRight: "7px solid  #232631" }}

      date={experience.date}

      iconStyle={{ background: experience.iconBg }}

      icon={

        <div className='flex justify-center items-center w-full h-full'>

          <img

            src={experience.icon}

            alt={experience.company\_name}

            className='w-[60%] h-[60%] object-contain'

          />

        </div>

      }

    >

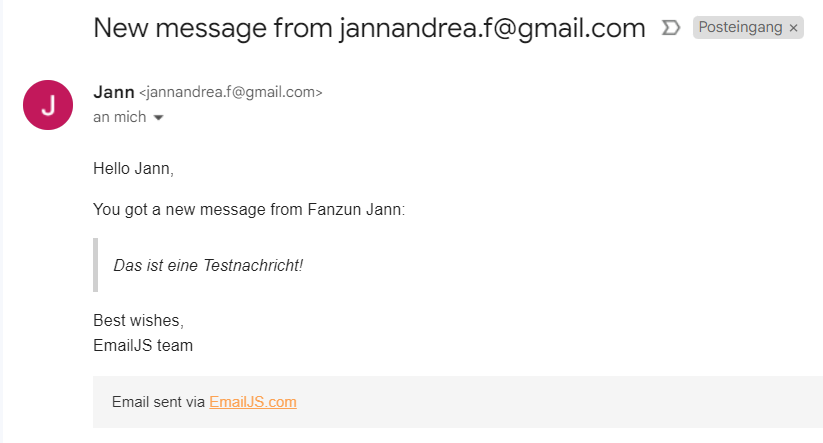
### Contact

Ich habe folgendes Vorgehen verwendet, um die Sicherheit meines Kontaktformulars zu gewährleisten, Nachrichten an mich zu senden und Daten mit EmailJS auszulesen:

* Import der erforderlichen Module und Bibliotheken.
* Initialisierung des Formularzustands mit useState.
* Implementierung des Handlers zum Ändern der Formulardaten.
* Implementierung des Handlers zum Absenden des Formulars mit EmailJS.
* Hinzufügen einer einfachen E-Mail-Validierung.
* Verwendung der Zustands- und Event-Handler-Funktionen im JSX.
* Anzeige einer Erfolgs- oder Fehlermeldung für den Benutzer.

Durch dieses Vorgehen konnte ich die Sicherheit gewährleisten und die Funktionalität des Kontaktformulars erfolgreich umsetzen.

Da ich die kostenlose Version von EmailJS verwendet habe, kann ich maximal 200 Nachrichten durch das Kontaktformular erhalten. Wenn ich eine Nachricht erhalte, sieht es folgendermassen aus:



20 Nachricht von EmailJS

* Im Header erkennt man die eingegebene Mail-Adresse
* In der Nachricht sieht man auf der zweiten Zeile den Namen
* In der dritten Zeile sieht man, was der User in das Kontaktformular geschrieben hat

### SectionWrapper

Der Code definiert eine Higher-Order Component (HOC) namens SectionWrapper, die einen animierten <motion.section>-Container rendert. Die wichtigsten Merkmale sind:

* Jede Section das gleiche Padding aufweist, unabhängig von der Bildschirmgrösse.
* Jede Section über eine eindeutige ID erreichbar ist.
* Animationen in den Sections integriert und verarbeitet werden können.

Die Funktion des Codes besteht darin, einen animierten Container für eine spezifische Sektion der Webseite zu erstellen, der Animationen und Übergänge ermöglicht und durch das Hinzufügen der ID die Navigation zu dieser Sektion erleichtert.

const SectionWrapper = (Component, idName) =>

function HOC () { return (

        <motion.section

            variants={staggerContainer()}

            initial="hidden"

            whileInView="show"

            viewport={{once: true, amount: 0.25}}

            className={`${styles.padding} max-w-7xl mx-auto relative z-0`}

        >

            <span className='hash-span' id={idName}>

                &nbsp;

            </span>

            <Component />

        </motion.section>

    )}

### CanvasLoader

Der ***Loader***-Code hat mir geholfen, eine Ladeanzeige für meinen Canvas zu implementieren. Dadurch wird verhindert, dass meine Seite beim Starten einfriert, da Benutzer den Fortschritt des Canvas-Ladevorgangs sehen können. Das verbessert die Benutzererfahrung und sorgt dafür, dass die Seite reibungslos geladen wird.



21 CanvasLoader

* Die Loader-Komponente rendert eine Ladeanzeige für den Fortschritt der 3D-Anwendung.
* Die useProgress-Hook wird verwendet, um den Fortschritt der Anwendung zu verfolgen und den aktuellen Wert in der progress-Variable zu erhalten.
* Die Ladeanzeige wird in einem <Html>-Element platziert.

const Loader = () => { const {progress} = useProgress();

  return (

    <Html>

      <span className="canvas-load flex justify-center text-center"></span>

      <p

      style={{

        fontSize: 14,

        color: '#f1f1f1',

        fontWeight: 800,

        marginTop: 40

      }}

      >{progress.toFixed(2)}%</p>

    </Html>

  )}

### Stern-Background

Der Code definiert die Funktion Stars, die Sterne in einer 3D-Darstellung rendert. Dieser Code hilft mir, meine Webseite mehr zum Leben zu bringen. Doch was macht der Code genau?

* Eine Referenz ref wird erstellt, um auf ein bestimmtes Element zuzugreifen.
* Der Zustand sphere wird mit useState definiert und enthält die Positionen der Sterne innerhalb einer Kugel.
* Die Funktion useFrame wird verwendet, um die Animation der Sterne zu steuern.
* In jedem Frame wird die Rotation des Elements, auf das ref zeigt, um kleine Werte verändert, um einen animierten Effekt zu erzeugen.
* Die Sterne drehen sich langsam um die X- und Y-Achse.

const Stars = (props) => {

  const ref = useRef();

  const [sphere] = useState(() => random.inSphere(new Float32Array(5001), { radius: 1.2 }));

  useFrame((state, delta) => {

    ref.current.rotation.x -= delta / 10;

    ref.current.rotation.y -= delta / 15;

  });

Zusammengefasst rendert der Code animierte Sterne in einer 3D-Darstellung und animiert sie durch Rotation um die X- und Y-Achse. Anschliessend werden die Sterne als Punkte mit bestimmten Positionen, Farbe und Größe dargestellt und in einer Gruppe platziert, die um 45 Grad um die Z-Achse gedreht ist.

## Daten und Content

Ich habe meine Daten und den gesamten Inhalt, wie Icons, Bilder, Texte, Links usw., strukturiert in JavaScript-Dateien exportiert. Dadurch konnte ich meinen Code so einfach wie möglich gestalten, indem ich die benötigten Inhalte importiere und mit den entsprechenden Parametern abrufe. Dadurch habe ich eine Effizienz erreicht, die es mir ermöglicht, immer weitere Texte, Karten usw. in eine JavaScript-Datei zu schreiben. Dadurch wird der Inhalt automatisch korrekt skaliert und zur Webseite hinzugefügt.

Meine Vorteile von diesem Vorgehen sind folgende:

* Vereinfachte Code-Struktur und verbesserte Lesbarkeit.
* Wiederverwendbarkeit der Daten in verschiedenen Teilen der Anwendung.
* Leichte Aktualisierung der Daten ohne Änderung des Hauptcodes.
* Einfaches Hinzufügen von weiteren Inhalten durch Verlinkung von Parametern.
* Minimierung potenzieller Fehler durch separate Validierung der Daten.
* Effiziente Zusammenarbeit zwischen Entwicklern.
* Erhöhte Wartbarkeit durch strukturierte Organisation der Daten.

## Deployment

Beim Deployment hatte ich zu Beginn Probleme. Als erstes habe ich geplant, mein Deployment über Github Pages zu tätigen, jedoch hatte ich da einige Probleme.

### Probleme

Fehlermeldung "***Failed to load resource: the server responded with a status of 404 ()***" in meiner Konsole. Meine Webseite wurde nicht angezeigt, nur ein weisser Bildschirm oder das README.

Mögliche Ursachen:

* Falsche Pfadangaben zu den Ressourcen im Code.
* Fehlerhafte Gross- und Kleinschreibung der Dateinamen.
* Fehlende Ressourcen oder falsche Dateistruktur im "dist"-Ordner.
* Nicht veröffentlichte Dateien im GitHub-Repository.

Mögliche Lösungen:

* Pfadangaben in der Anwendung überprüfen.
* Gross- und Kleinschreibung der Dateinamen überprüfen.
* Dateistruktur im "dist"-Ordner überprüfen.
* Sicherstellen, dass alle Dateien im Repository veröffentlicht sind.
* "gh-pages"-Branch löschen und erneut deployen.

### Lösung

Als ich die Probleme nicht richtig lösen konnte, habe ich ein anderes Tool kennengelernt, um mein Deployment durchzuführen. Vercel hat mir geholfen, meine Webseite einfach zu deployen und hochzuladen.

Vorteile des Deployments eines GitHub-Repositorys auf Vercel:

* Einfacher und schneller Deploy-Prozess.
* Automatische Integration mit GitHub-Repositories.
* Kostenlose Option für kleinere Projekte.
* Unterstützung für verschiedene Frameworks und Sprachen.
* Skalierbarkeit und hohe Performance durch Vercels Serverless-Architektur.

Ablauf des Deployments eines GitHub-Repositorys auf Vercel:

1. Erstellen eines Vercel-Kontos und Anmelden.
2. Erstellen eines neuen Projekts in Vercel.
3. Bei der Projektkonfiguration die Option "Import Git Repository" wählen.
4. Das gewünschte GitHub-Repository auswählen.
5. Einstellungen wie Branch, Verzeichnis und Build-Konfiguration festlegen.
6. Vercel führt automatisch einen Build durch und deployed die Anwendung.
7. Bereitstellung der öffentlichen URL für die deployed Anwendung.
8. Automatische Aktualisierung bei jedem Push auf den ausgewählten Branch im GitHub-Repository.

Durch die Nutzung von Vercel zur Bereitstellung meines GitHub-Repository konnte ich den Deploy-Prozess vereinfachen und meine Anwendung schnell und zuverlässig veröffentlichen.

# Kontrollieren

## Testfall Figuren bewegen

|  |  |
| --- | --- |
| **Testfall Nr.** | **1** |
| Beschreibung | mit linker Maustaste drehen – mit rechter Maustaste bewegen |
| Voraussetzungen | 3d Form unberührt und im normalen Zustand |
| Testschritte | * Form drehen * Form bewegen |
| Erwartetes Ergebnis | Form kann interaktiv bewegt und gedreht werden |
| Testergebnis | erfüllt |
| Testdatum | 26.06.2023 |
| Bemerkungen | - |

## Testfall Links öffnen

|  |  |
| --- | --- |
| **Testfall Nr.** | **1** |
| Beschreibung | mit Klick auf Link wird Link geöffnet |
| Voraussetzungen | Link wurde noch nicht geöffnet |
| Testschritte | * Klick auf Link * Link wird geöffnet und URL wird angezeigt |
| Erwartetes Ergebnis | neues Fenster geht auf mit erwünschten URL |
| Testergebnis | erfüllt |
| Testdatum | 26.06.2023 |
| Bemerkungen | - |

## Testfälle Nachricht senden

|  |  |
| --- | --- |
| **Testfall Nr.** | **1** |
| Beschreibung | in Textfeld wird etwas geschrieben und Nachricht wird versendet |
| Voraussetzungen | Textfelder sind leer |
| Testschritte | * In Textfeld kann geschrieben werden |
| Erwartetes Ergebnis | Man kann in Textfelder schreiben und sie leeren sich |
| Testergebnis | erfüllt |
| Testdatum | 26.06.2023 |
| Bemerkungen | - |

|  |  |
| --- | --- |
| **Testfall Nr.** | **2** |
| Beschreibung | E-Mail-Syntax wird überprüft |
| Voraussetzungen | falsche E-Mail eingeben (ohne @ oder .xxx) |
| Testschritte | * Falscher Syntax wird eingegeben * Nach «Send» wird Feld geleert |
| Erwartetes Ergebnis | Fehlermeldung wird gegeben – Nachricht kann nicht gesendet werden |
| Testergebnis | nicht erfüllt |
| Testdatum | 26.06.2023 |
| Bemerkungen | ich werde es anpassen |

|  |  |
| --- | --- |
| **Testfall Nr.** | **3** |
| Beschreibung | Nachricht senden ohne Nachricht |
| Voraussetzungen | Nachrichtfeld bleibt leer |
| Testschritte | * E-Mail und Name eingeben * Nachrichtfeld leer lassen |
| Erwartetes Ergebnis | Fehlermeldung, Nachricht kann nicht gesendet werden |
| Testergebnis | nicht erfüllt |
| Testdatum | 26.06.2023 |
| Bemerkungen | ich werde es anpassen |

## Testfall Navigation auf der Webseite

|  |  |
| --- | --- |
| **Testfall Nr.** | **1** |
| Beschreibung | Links in der Navigationsleiste funktionieren einwandfrei |
| Voraussetzungen | Links werden angezeigt und können geklickt werden |
| Testschritte | * Auf jeden Link drücken |
| Erwartetes Ergebnis | Navigation führt zum gewollten Punkt |
| Testergebnis | erfüllt |
| Testdatum | 26.06.2023 |
| Bemerkungen | - |

# Auswerten

Meine Auswertung zu meiner Webseite mit Three.js, React und Tailwind CSS Das Hauptziel des Projekts bestand darin, eine persönliche Portfolio-Website zu erstellen, auf der ich meine Fähigkeiten und bisherigen Projekte präsentieren kann. Hier ist eine strukturierte und einfache Zusammenfassung meiner Erfolge:

**Erreichte Ziele:**

a) Persönliche Webseite: Ich habe erfolgreich eine Webseite entwickelt, auf der ich meine Fähigkeiten und Projekte präsentieren kann. Die Webseite bietet einen intuitiven und benutzerfreundlichen Zugang zu den Inhalten.

b) Neue Programmiersprache: Durch die Entwicklung dieser Webseite habe ich meine Kenntnisse in Three.js (React) erweitert. Ich habe mich erfolgreich in die Sprache eingearbeitet und praktische Erfahrungen gesammelt.

c) Fortschrittsverfolgung: Die Webseite dient auch dazu, meine Fortschritte und Fähigkeiten als Entwickler festzuhalten. Ich plane, regelmäßig über meine Lernerfolge, Herausforderungen und neue Fähigkeiten zu berichten.

**Webdesign:**

Die Webseite wurde mit einem modernen und ästhetisch ansprechenden Design gestaltet. Das Ziel war es, den Besuchern ein positives Nutzererlebnis zu bieten. Dank der Verwendung von Three.js, React und Tailwind CSS konnte ich das gewünschte Design erfolgreich umsetzen.

**Kundenmeinungen / Testimonials:**

Ich habe dieses Ziel fast erreicht. Ich habe begonnen aber die Kundenmeinungen-Sektion nicht ganz abgeschlossen.

**Kontaktformular:**

Obwohl ich beim Testen festgestellt habe, dass das Kontaktformular noch Verbesserungen in Bezug auf Security und Bug-Fixes erfordert, konnte ich dennoch ein funktionierendes Kontaktformular implementieren. Es bietet potenziellen Interessenten eine einfache Möglichkeit, mich zu kontaktieren.

**Responsives Design:**

Die Webseite wurde so entwickelt, dass sie auf verschiedenen Geräten und Bildschirmgrößen gut funktioniert. Dank des responsiven Designs können Besucher die Webseite problemlos auf Mobilgeräten, Tablets und Desktops anzeigen.

Insgesamt bin ich mit den Ergebnissen des Projekts zufrieden. Ich habe fast alle meine Ziele erreicht und konnte meine Kenntnisse in Three.js (React), Tailwind CSS und Webdesign erweitern. Das Projekt hat mir die Möglichkeit gegeben, meine Fähigkeiten als Entwickler zu präsentieren und mein Portfolio zu verbessern. Für zukünftige Arbeiten werde ich mich auf die Verbesserung der Security und die Behebung von Bugs im Kontaktformular konzentrieren.

# Quellenverzeichnis

|  |  |
| --- | --- |
| Link | Beschreibung / Datum |
| <https://youtu.be/GOuwOI-WSkE> | Framer Motion / Juni 2023 |
| <https://youtu.be/9ZEjSxDRIik> | 3D Three JS Fiber / Juni 2023 |
| <https://youtu.be/Q7AOvWpIVHU> | 3D Portfolio Inspiration / Juni 2023 |
| [https://sketchfab.com/3D-models/gaming-desktop-pc-d1d8282c9916438091f11aeb28787b66](https://sketchfab.com/3d-models/gaming-desktop-pc-d1d8282c9916438091f11aeb28787b66) | 3D Models / Juni 2023 |
| <https://logo.com/> | Logo creator / Juni 2023 |
| <https://www.emailjs.com/> | EmailJS / Juni 2023 |
| <https://docs.pmnd.rs/react-three-fiber/getting-started/introduction> | React Three Fiber Doc / Juni 2023 |
| <https://www.framer.com/motion/> | Framer Motion Doc / Juni 2023 |
| <https://fonts.googleapis.com/css2?family=Poppins:wght@100;200;300;400;500;600;700;800;900&display=swap> | Google Fonts / Juni 2023 |
| <https://youtu.be/Q7AOvWpIVHU> | 3D Webseite Inspiration / Juni 2023 |
| <https://youtu.be/0fYi8SGA20k> | 3D Webseite Inspiration / Juni 2023 |
| <https://youtu.be/qALsVa-V9qo> | 3D Webseite Inspiration / Juni 2023 |
| <https://youtu.be/pggIVY5eOGM> | Fusszeile in React / Juni 2023 |
| [https://sketchfab.com/3D-models/earth-hologram-87072288fb234226b9a3f02ae674a310](https://sketchfab.com/3d-models/earth-hologram-87072288fb234226b9a3f02ae674a310) | 3D Models / Juni 2023 |
| <https://tailwindcss.com/docs/installation> | Tailwind CSS / Juni 2023 |

# Abbildungsverzeichnis

[1 Gantt Diagramm 10](file:///C:\Users\jfanzun\Documents\0_4_ZLI\3_0_Projekte\9_0_Abschlussarbeit\Dokumente\Projektdokumentation_3d_Portfolio_Abschlussarbeit_ZLI.docx#_Toc138774231)

[2 Kanban Board 11](file:///C:\Users\jfanzun\Documents\0_4_ZLI\3_0_Projekte\9_0_Abschlussarbeit\Dokumente\Projektdokumentation_3d_Portfolio_Abschlussarbeit_ZLI.docx#_Toc138774232)

[3 Wireframe Laptop 12](file:///C:\Users\jfanzun\Documents\0_4_ZLI\3_0_Projekte\9_0_Abschlussarbeit\Dokumente\Projektdokumentation_3d_Portfolio_Abschlussarbeit_ZLI.docx#_Toc138774233)

[4 Wireframe Telefon 13](file:///C:\Users\jfanzun\Documents\0_4_ZLI\3_0_Projekte\9_0_Abschlussarbeit\Dokumente\Projektdokumentation_3d_Portfolio_Abschlussarbeit_ZLI.docx#_Toc138774234)

[5 Flow Chart 14](file:///C:\Users\jfanzun\Documents\0_4_ZLI\3_0_Projekte\9_0_Abschlussarbeit\Dokumente\Projektdokumentation_3d_Portfolio_Abschlussarbeit_ZLI.docx#_Toc138774235)

[6 Struktur src 15](file:///C:\Users\jfanzun\Documents\0_4_ZLI\3_0_Projekte\9_0_Abschlussarbeit\Dokumente\Projektdokumentation_3d_Portfolio_Abschlussarbeit_ZLI.docx#_Toc138774236)

[7 Struktur public 15](file:///C:\Users\jfanzun\Documents\0_4_ZLI\3_0_Projekte\9_0_Abschlussarbeit\Dokumente\Projektdokumentation_3d_Portfolio_Abschlussarbeit_ZLI.docx#_Toc138774237)

[8 potenzielle Logos und Favicons 16](file:///C:\Users\jfanzun\Documents\0_4_ZLI\3_0_Projekte\9_0_Abschlussarbeit\Dokumente\Projektdokumentation_3d_Portfolio_Abschlussarbeit_ZLI.docx#_Toc138774238)

[9 PC 3d Modelle 17](file:///C:\Users\jfanzun\Documents\0_4_ZLI\3_0_Projekte\9_0_Abschlussarbeit\Dokumente\Projektdokumentation_3d_Portfolio_Abschlussarbeit_ZLI.docx#_Toc138774239)

[10 Weltkugel 3d Modelle 17](file:///C:\Users\jfanzun\Documents\0_4_ZLI\3_0_Projekte\9_0_Abschlussarbeit\Dokumente\Projektdokumentation_3d_Portfolio_Abschlussarbeit_ZLI.docx#_Toc138774240)

[11 definitive 3d Figuren 18](file:///C:\Users\jfanzun\Documents\0_4_ZLI\3_0_Projekte\9_0_Abschlussarbeit\Dokumente\Projektdokumentation_3d_Portfolio_Abschlussarbeit_ZLI.docx#_Toc138774241)

[12 definitive Logos 19](file:///C:\Users\jfanzun\Documents\0_4_ZLI\3_0_Projekte\9_0_Abschlussarbeit\Dokumente\Projektdokumentation_3d_Portfolio_Abschlussarbeit_ZLI.docx#_Toc138774242)

[13 Ordner & Files nach Initialisierung 22](file:///C:\Users\jfanzun\Documents\0_4_ZLI\3_0_Projekte\9_0_Abschlussarbeit\Dokumente\Projektdokumentation_3d_Portfolio_Abschlussarbeit_ZLI.docx#_Toc138774243)

[14 Navigationsleiste 23](file:///C:\Users\jfanzun\Documents\0_4_ZLI\3_0_Projekte\9_0_Abschlussarbeit\Dokumente\Projektdokumentation_3d_Portfolio_Abschlussarbeit_ZLI.docx#_Toc138774244)

[15 Entwicklung der Navigationsleiste 23](file:///C:\Users\jfanzun\Documents\0_4_ZLI\3_0_Projekte\9_0_Abschlussarbeit\Dokumente\Projektdokumentation_3d_Portfolio_Abschlussarbeit_ZLI.docx#_Toc138774245)

[16 3D Modell 24](file:///C:\Users\jfanzun\Documents\0_4_ZLI\3_0_Projekte\9_0_Abschlussarbeit\Dokumente\Projektdokumentation_3d_Portfolio_Abschlussarbeit_ZLI.docx#_Toc138774246)

[19 Work Abschnitt 26](file:///C:\Users\jfanzun\Documents\0_4_ZLI\3_0_Projekte\9_0_Abschlussarbeit\Dokumente\Projektdokumentation_3d_Portfolio_Abschlussarbeit_ZLI.docx#_Toc138774247)

[20 Nachricht von EmailJS 27](file:///C:\Users\jfanzun\Documents\0_4_ZLI\3_0_Projekte\9_0_Abschlussarbeit\Dokumente\Projektdokumentation_3d_Portfolio_Abschlussarbeit_ZLI.docx#_Toc138774248)

[21 CanvasLoader 28](file:///C:\Users\jfanzun\Documents\0_4_ZLI\3_0_Projekte\9_0_Abschlussarbeit\Dokumente\Projektdokumentation_3d_Portfolio_Abschlussarbeit_ZLI.docx#_Toc138774249)