

Event-Planer

Felix Hoffmann, Baran Bickici, Sami Gökpinar, Ergün Bickici

30. März 2025

Inhaltsverzeichnis

1	Einführung	3
2	Technologien	3
2.1	Full-Stack Framework	3
2.1.1	Frontend	3
2.1.2	Backend	3
2.2	Datenbank	4
2.3	Authentifizierung	4
2.4	Deliverable	4
3	Marktanalyse	5
3.1	Zielgruppe	5
3.2	Bewertung ähnlicher Produkte	5
4	Requirements Specification	6
4.1	Kundenbefragung	6
4.2	User stories	6
4.3	Personas	6
5	Funktionsumfang	7
5.1	Anmeldung	7
6	UI Entwürfe	8
6.1	Events	8
6.1.1	Erstellen	8
6.1.2	Bearbeiten	8
6.1.3	Löschen	8
6.1.4	Teilnahme	8
6.2	Wishes	8
6.2.1	Erstellen	8
6.2.2	Bearbeiten	8
6.2.3	Löschen	8
6.2.4	Upvote	8
7	UX-Optimierung	9
7.1	Gründe für eine UX-Optimierung	9
7.2	Worauf kommt es bei der UX-Optimierung an?	9
7.3	Benutzerfreundlichkeit	9
7.4	Barrierefreiheit	9
8	Aufwandsschätzung	10
8.1	Roadmap	10
9	Projektmanagement: "Scrum-ban"	11
10	Systemarchitektur	12
11	Literaturverzeichnis	13

1 Einführung

Im Rahmen des Softwareprojektes im 4. Semester des Studienganges Softwaretechnik und Medieninformatik wird über das ganze Semester ein Projekt mit einer Partnerfirma, die als Kunden agieren durchgeführt. Dieses Projekt wird mit der Firma Pep-Digital die sich ein Produkt wünschen mit dem Titel "Event-Planer". Dieses Produkt soll eine Web-Applikation werden, in denen Mitarbeiter der Firma Pep-Digital Events erstellen und das Event planen können z.B. mit Umfragen, Datum des Events und Teilnehmer. Dazu hat man auch die Möglichkeit Wünsche zu äußern aus denen man Events erstellen kann.

2 Technologien

2.1 Full-Stack Framework

Für die Umsetzung des Projektes wurde Next.js als Full-Stack-Framework gewählt. Next.js basiert auf React und bietet eine vollständige Lösung für die Entwicklung von Webanwendungen, indem es sowohl Frontend- als auch Backend-Funktionalitäten integriert. Mit Next.js können Entwickler sowohl serverseitiges Rendering (SSR) als auch statische Seitengenerierung (SSG) nutzen. Zudem bietet es eine einfache Möglichkeit, API-Routen zu erstellen, was es zu einer hervorragenden Wahl für Full-Stack-Entwicklungen macht. Die Verwendung von TypeScript sorgt für eine typsichere und fehlerarme Entwicklung, sowohl im Frontend als auch im Backend.

2.1.1 Frontend

Im Frontend nutzt Next.js React. React ist eine quelloffene JavaScript-Bibliothek, die das Erstellen der Benutzeroberfläche schnell und dynamisch macht. Anhand von React können Web- und Mobile-Anwendungen mit derselben Codebasis erstellt werden, ohne jegliche Formatierung. Die Codierung erfolgt in TypeScript, was durch statische Typisierung und bessere Fehlererkennung das Entwickeln sicherer und effizienter macht.

2.1.2 Backend

Das Backend in Next.js wird durch das integrierte API-Routing realisiert, das auf Node.js basiert. Durch diese Integration können Entwickler API-Routen direkt innerhalb der Next.js-Anwendung erstellen, ohne einen separaten Server benötigen zu müssen. Diese Routen ermöglichen es, serverseitige Logik auszuführen, wie etwa das Abrufen von Daten aus einer Datenbank oder das Bearbeiten von Anfragen. Da Next.js auf Node.js aufbaut, können alle leistungsfähigen Funktionen von Node.js genutzt werden, während TypeScript die Entwicklung durch statische Typisierung und Fehlererkennung verbessert. Dies sorgt für eine konsistente und sichere Entwicklung sowohl im Frontend als auch im Backend innerhalb derselben Codebasis.

2.2 Datenbank

Für die Datenbank wurde PostgreSQL in Kombination mit Supabase gewählt. PostgreSQL bietet zahlreiche Vorteile für die Nutzung in Webanwendungen. Es ist besonders leistungsfähig und skalierbar, was es ideal für die Verwaltung großer Datenmengen und viele gleichzeitige Nutzer macht. Mit seiner hohen Erweiterbarkeit ermöglicht es die Anpassung der Datenbank an spezifische Anforderungen, etwa durch benutzerdefinierte Datentypen und Funktionen. Die robuste Datenintegrität sorgt dafür, dass die Daten sicher und konsistent bleiben, auch bei komplexen Transaktionen. Dank fortschrittlicher Indexierungs- und Abfrageoptimierungen können Webanwendungen schnell auf große Datenmengen zugreifen. Zudem ist PostgreSQL ACID-konform, was bedeutet, dass es zuverlässige und fehlerfreie Transaktionen garantiert. Diese Eigenschaften machen PostgreSQL zu einer bevorzugten Wahl für Webanwendungen, die hohe Leistung, Flexibilität und Datenintegrität erfordern. Supabase ist eine Open-Source-Plattform, die Entwicklern hilft, moderne Anwendungen schnell und einfach zu erstellen. Sie bietet eine Vielzahl von Funktionen wie Datenbanken, Authentifizierung, Echtzeit-Updates und APIs, um skalierbare und sichere Anwendungen zu entwickeln, ohne viel Aufwand in die Backend-Programmierung investieren zu müssen.

2.3 Authentifizierung

Supabase wird für die Authentifizierung genutzt, da es eine einfache und sichere Möglichkeit bietet, Nutzer in eine Anwendung zu integrieren. Es unterstützt Single Sign-On (SSO) und externe Identitätsanbieter wie Google und Microsoft Azure, sodass sich Benutzer bequem mit ihren bestehenden Konten anmelden können. Durch die Integration von Microsoft Azure können sich insbesondere Mitarbeiter direkt mit ihren Microsoft-Accounts authentifizieren. Dies erleichtert den Zugriff auf die Anwendung, da keine separaten Anmeldeinformationen erstellt werden müssen, und sorgt gleichzeitig für höhere Sicherheit und bessere Verwaltungsmöglichkeiten durch zentrale Benutzerkontrollen in Azure.

2.4 Deliverable

Als Deliverable wird in diesem Projekt Docker verwendet, da so sicher gegangen werden kann, dass die Anwendung in einer konsistenten Umgebung ausgeführt wird, unabhängig von den zugrunde liegenden Betriebssystemen oder Hardwarekonfigurationen.

3 Marktanalyse

3.1 Zielgruppe

3.2 Bewertung ähnlicher Produkte

4 Requirements Specification

4.1 Kundenbefragung

4.2 User stories

4.3 Personas

5 Funktionsumfang

5.1 Anmeldung

erkläre funktion der app

6 UI Entwürfe

6.1 Events

6.1.1 Erstellen

6.1.2 Bearbeiten

6.1.3 Löschen

6.1.4 Teilnahme

6.2 Wishes

6.2.1 Erstellen

6.2.2 Bearbeiten

6.2.3 Löschen

6.2.4 Upvote

7 UX-Optimierung

7.1 Gründe für eine UX-Optimierung

7.2 Worauf kommt es bei der UX-Optimierung an?

7.3 Benutzerfreundlichkeit

7.4 Barrierefreiheit

8 Aufwandsschätzung

8.1 Roadmap

9 Projektmanagement: "Scrum-ban"

10 Systemarchitektur

11 Literaturverzeichnis