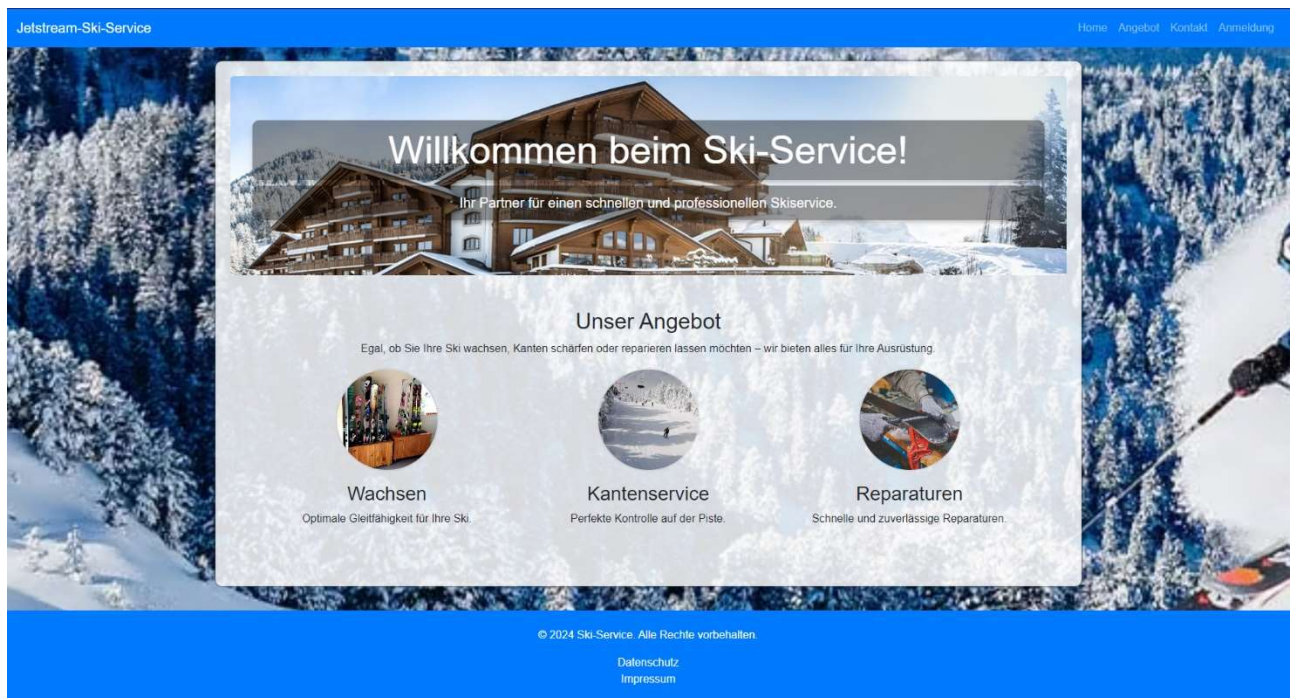


# IPERKA-Dokumentation zum Ski-Service-Projekt



Von Tunahan, Yannick & Felipe

## Inhalt

1. Informieren .....	3
Ziel des Projekts .....	3
Rahmenbedingungen .....	3
2. Planen .....	4
Projektstruktur .....	4
Projektphasen .....	4
3. Entscheiden .....	5
Wichtige Entscheidungen für das Projekt.....	5
4. Realisieren.....	6
Umsetzungsschritte .....	6
Herausforderungen: .....	7
5. Kontrollieren .....	7
1. API-Endpunkt-Tests: .....	7
2. Integrationstests: .....	7
6. Auswerten.....	8
Erfolgskriterien .....	8

# 1. Informieren

## Ziel des Projekts

Ziel des Projekts ist die Entwicklung eines Skidienst-Backends, das verschiedene API-Funktionen bietet.

## Rahmenbedingungen

- Technologie: Node.js (Backend).
- Bibliotheken: Express für Routing, dotenv für Umgebungsvariablen, cors für Cross-Origin Requests.
- Tools: Swagger zur API-Dokumentation, Nodemon zur Entwicklungsunterstützung.
- Voraussetzungen: Installation von Node.js und den aufgeführten Bibliotheken.



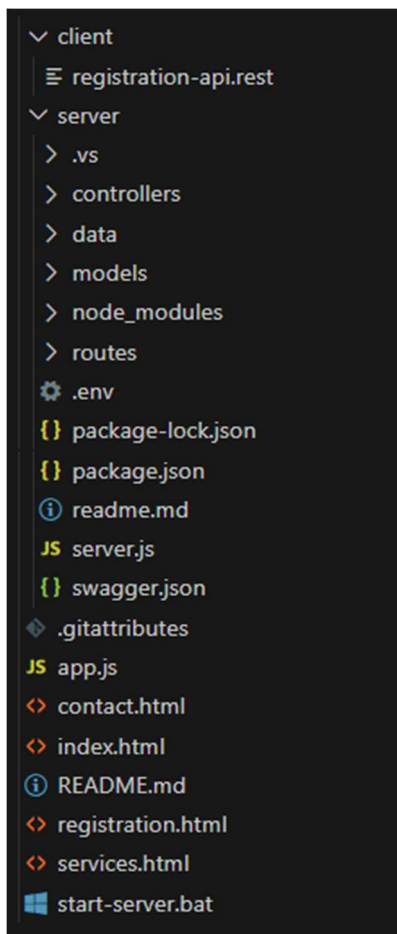
## 2. Planen

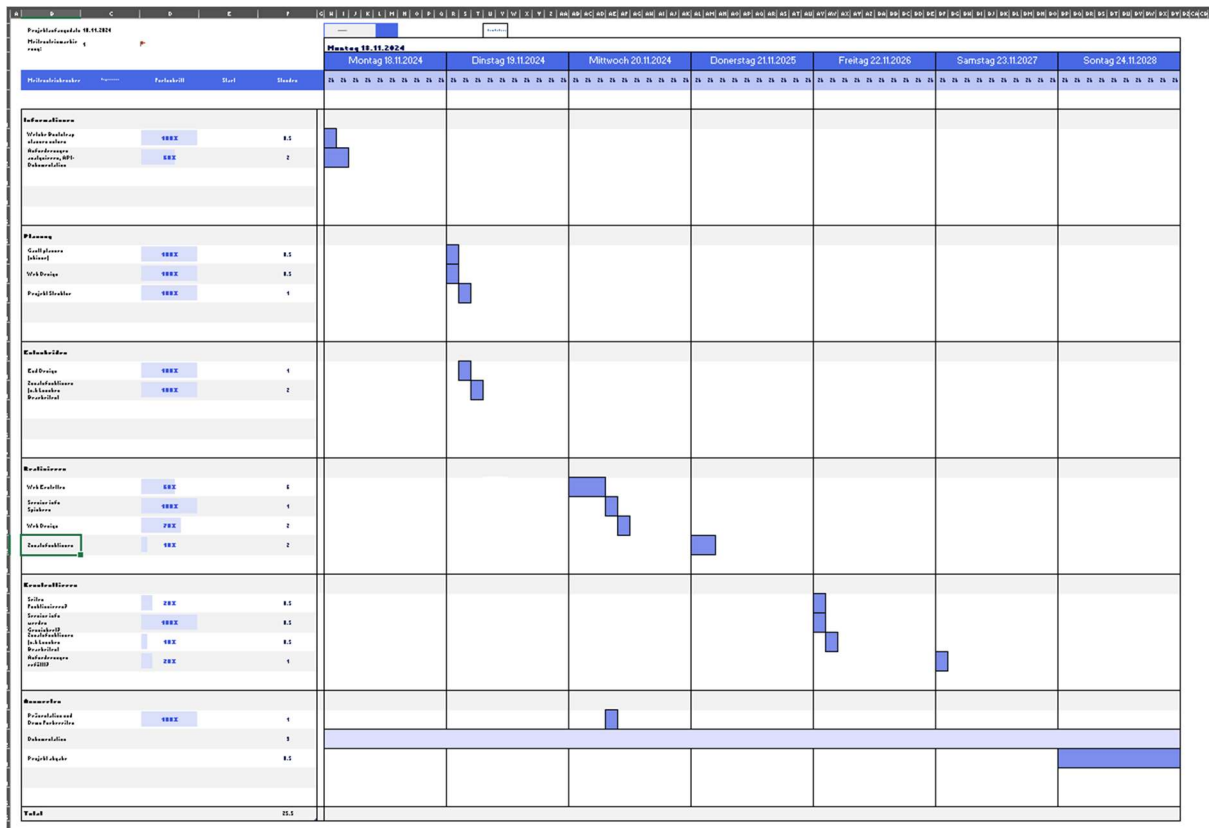
### Projektstruktur

- Server: Enthält den Backend-Code.
- Client und Ressourcen: Frontend-Dateien liegen in den Verzeichnissen client, html, css, js.
- Zusätzliche Dokumente:
  - Präsentation (modul294 präsi.pptx) zur Unterstützung.
  - Gantt-Diagramm (Gantt-Diagramm-ski.xlsx) zur zeitlichen Planung.

### Projektphasen

- Einrichtung der Entwicklungsumgebung (Node.js und Bibliotheken).
- Implementierung der API mit Express.
- Testen und Debugging mit Nodemon.
- Bereitstellung und Präsentation.





### Zeitplanung (siehe Gantt-Diagramm):

- **18.11.2024:** Einrichtung der Entwicklungsumgebung (Node.js, Bibliotheken, Verzeichnisstruktur).
- **19.11.2024:** Entwicklung der API (CRUD-Funktionen, Routing, Swagger-Integration).
- **20.11.2024:** Testen und Debugging der API, Fehlerbehebung und Optimierung.
- **21.11.2024:** Abschlussarbeiten: Dokumentation, Präsentation und finaler Testlauf.

## 3. Entscheiden

### Wichtige Entscheidungen für das Projekt

Während der Planungsphase wurden mehrere entscheidende Punkte besprochen und festgelegt, die den Erfolg des Projekts sicherstellen sollen. Diese Entscheidungen umfassen sowohl technologische als auch organisatorische Aspekte:

#### 1. Technologieauswahl:

- **Node.js als backend:**  
Node.js wurde aufgrund seiner hohen Performance und Flexibilität ausgewählt. Es ermöglicht eine einfache Entwicklung und Integration mit dem Frontend sowie eine große Anzahl unterstützender Bibliotheken.
- **Express als Framework:**  
Express wurde gewählt, da es eine schlanke und effiziente Möglichkeit bietet, APIs zu erstellen. Es vereinfacht die Verwaltung von Routen und Middleware.

## 2. Sicherheitsmaßnahmen:

- cors-Integration:  
Um eine sichere Kommunikation zwischen dem Frontend und backend zu gewährleisten, wurde die Integration von CORS beschlossen. Dies löst typische Probleme bei Cross-Origin-Anfragen und schützt die API vor unautorisierten Zugriffen.
- dotenv für Umgebungsvariablen:  
Die Verwendung von dotenv ermöglicht es, sensible Informationen wie API-Schlüssel oder Server-Ports sicher in einer .env-Datei zu speichern. Dadurch bleibt der Quellcode sauber und sicher.

## 3. Projektorganisation:

- Zeitplanung:  
Die Erstellung eines Gantt-Diagramms wurde beschlossen, um den Projektverlauf detailliert zu planen. Dies ermöglicht eine klare Übersicht über Deadlines und eine optimale Ressourcenzuweisung.
- Teamaufteilung:  
Aufgaben wurden klar zwischen Tunahan, Yannick und Felipe verteilt, um die Effizienz zu maximieren und Überlappungen zu vermeiden. Jede Phase wurde einem Verantwortlichen zugeteilt, wobei regelmäßige Abstimmungen sicherstellen, dass alle Teammitglieder auf dem gleichen Stand bleiben.

## 4. Teststrategie:

- Um sicherzustellen, dass das Backend fehlerfrei arbeitet, wurden umfassende Tests als integraler Bestandteil des Projekts festgelegt. Die Entscheidung fiel auf ein schrittweises Testverfahren: Funktionstests für die API-Endpunkte, Integrationstests mit dem Frontend und abschließende Systemtests.

# 4. Realisieren

Die praktische Umsetzung des Projekts folgte einem klaren, strukturierten Prozess, um sicherzustellen, dass alle Anforderungen termingerecht und in hoher Qualität erfüllt wurden. Die Schritte umfassen die Einrichtung der Umgebung, die Entwicklung der API, die Dokumentation, das Testen und abschließende Verbesserungen.

## Umsetzungsschritte

1. Setup: Initialisierung des Projekts mit npm init und Installation der Bibliotheken.
2. API-Entwicklung:
  - Definition der Endpunkte (z. B. CRUD-Operationen für Ski-Services).
  - Konfiguration von Umgebungsvariablen mit dotenv.

3. Testlauf: Starten des Servers mit npm run dev.

## Herausforderungen:

- Sicherstellen der Kompatibilität zwischen Backend und Frontend.
- Umsetzung der Sicherheitsrichtlinien (z. B. Cross-Origin Resource Sharing).

## 5. Kontrollieren

Die Kontrollphase ist entscheidend, um sicherzustellen, dass das entwickelte Backend die Anforderungen erfüllt und stabil arbeitet. In dieser Phase wurden detaillierte Tests durchgeführt, um die Funktionalität der API zu validieren, die Integration mit dem Frontend zu prüfen und die Qualität des gesamten Projekts sicherzustellen.

### Testszenarios und Methodik

#### 1. API-Endpunkt-Tests:

Jeder Endpunkt der API wurde auf Basis definierter Anforderungen getestet. Dies beinhaltete:

- GET-Anfragen: Überprüfung, ob Daten korrekt aus der Datenbank abgerufen werden.
- POST-Anfragen: Sicherstellen, dass neue Einträge in der Datenbank erstellt werden können.
- PUT-Anfragen: Überprüfung, ob bestehende Datensätze erfolgreich aktualisiert werden.
- DELETE-Anfragen: Test, ob Daten sicher und korrekt gelöscht werden.

#### Werkzeuge:

- Postman: Zum manuellen Testen der Endpunkte.

#### 2. Integrationstests:

Um die Zusammenarbeit zwischen Backend und Frontend zu prüfen, wurden folgende Szenarien getestet:

- Verbindungstests: Überprüfung, ob das Frontend erfolgreich auf die API-Endpunkte zugreifen kann.
- Datenfluss: Sicherstellen, dass die Daten korrekt zwischen Frontend und Backend übertragen werden.
- Fehlersimulation: Testen von Fehlerszenarien, z. B. ungültige Anfragen oder Verbindungsprobleme, um sicherzustellen, dass die API robuste Fehlermeldungen ausgibt.

#### Werkzeuge:

- Browser-Debugging: Zur Analyse der Frontend-API-Kommunikation.

## 6. Auswerten

Die Auswertung des Ski-Service-Projekts dient dazu, die Ergebnisse zu analysieren, die Erreichung der Ziele zu bewerten und Erkenntnisse für zukünftige Projekte zu gewinnen. Dabei wurden sowohl technische als auch organisatorische Aspekte berücksichtigt.

### Erfolgskriterien

Die folgenden Erfolgskriterien wurden vorab definiert, um den Projekterfolg zu messen:

#### **1. Funktionalität des Backends:**

Das Backend sollte vollständig funktionsfähig sein und alle geplanten CRUD-Operationen (Create, Read, Update, Delete) unterstützen.

#### **2. Dokumentation:**

Die API-Dokumentation musste klar, verständlich und vollständig sein, sodass Entwickler sie problemlos nutzen können.

#### **3. Integration mit dem Frontend:**

Die Verbindung zwischen Backend und Frontend sollte reibungslos funktionieren, um eine gute Benutzererfahrung zu gewährleisten.

#### **4. Termineinhaltung:**

Alle geplanten Aufgaben mussten im vorgegebenen Zeitrahmen abgeschlossen werden.

#### **5. Teamarbeit:**

Die Aufgabenverteilung im Team sollte effizient sein und die Zusammenarbeit sollte zu einem harmonischen Projektverlauf führen.