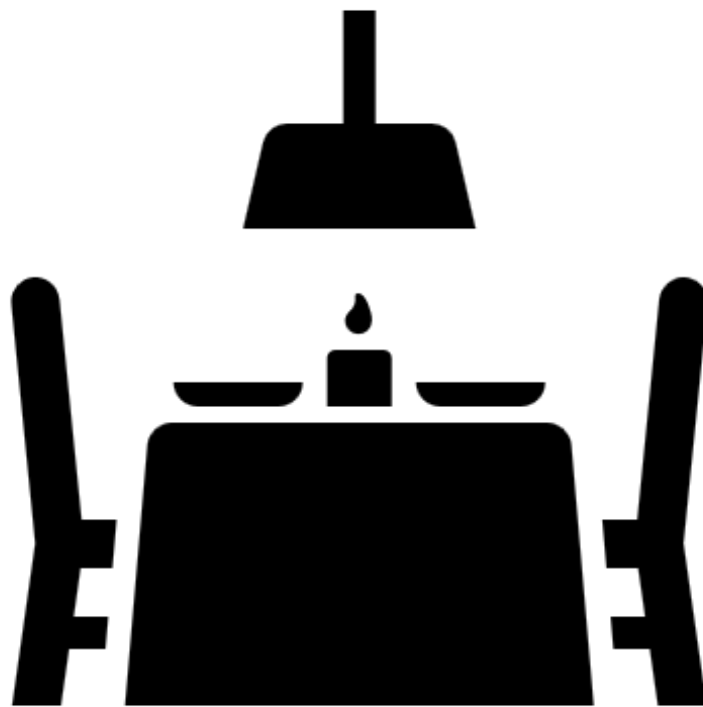


Webapplikation für ein Restaurant



Thema: Fullstack Applikation

Dokumentinformationen

Dateiname: 2025-rest-foods-7
Speicherdatum: 25.06.2025

Autoreninformationen

Autor: Diego De Corso, Luka Grujic, Nepomuk Crhonek, Sylia Scandinaro

E-Mail: diego.decorso@noseryoung.ch
luka.grujic@noseryoung.ch
nepomuk.crhonek@noseryoung.ch
sylia.scandinaro@noseryoung.ch

Inhaltsverzeichnis

1	Einleitung.....	3
2	Hauptteil.....	3
2.1	Informieren	3
2.2	Planen.....	4
2.2.1	Aufgabenplanung.....	4
2.2.2	Ablaufplanung	4
2.2.3	Frontendplanung.....	5
2.2.4	Programmstruktur und Aufgabenverteilung	6
2.3	Entscheiden.....	7
2.4	Realisieren.....	7
2.4.1	Frontend.....	7
2.4.2	Backend.....	7
2.5	Kontrollieren.....	8
2.5.1	Funktionsprüfung.....	8
2.5.2	API-Tests mit Postman	8
2.5.3	Fehlererkennung und Behebung	8
2.5.4	Stabilitäts- und Usability-Verbesserung	9
2.5.5	Fazit der Kontrollieren-Phase	9
2.6	Auswerten.....	9
2.6.1	Reflexion	9
3	Schluss	10

1 Einleitung

Das vorliegende Dokument beschreibt den Verlauf unseres Projekts zur Entwicklung einer Webapplikation für das Restaurant REST-Foods, welches demnächst einen neuen Standort an der Herostrasse 12 eröffnet. Da bisher keine Marketingmassnahmen umgesetzt wurden, wurde entschieden, eine moderne und benutzerfreundliche Website zu erstellen, die als zentraler digitaler Auftritt des Restaurants dient.

Ziel der Webapplikation ist es, interessierten Kunden umfassende Informationen zum Restaurant bereitzustellen – darunter die Speisekarte, Öffnungszeiten und Standortangaben. Zusätzlich soll eine Online-Reservierungsfunktion integriert werden, über die Gäste bequem einen Tisch buchen können.

In einem Dokument haben die Inhaber von REST-Foods die Anforderungen und Erwartungen an die Applikation definiert und in einem Vertrag festgehalten. Die vorliegende Dokumentation zeigt den gesamten Projektverlauf – von der Planung über die Umsetzung bis hin zur abschliessenden Auswertung.

2 Hauptteil

2.1 Informieren

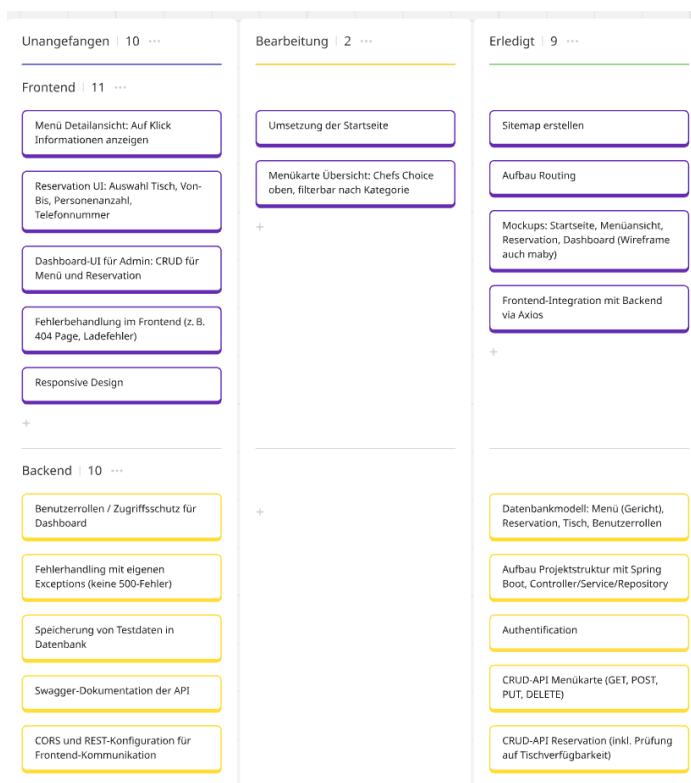
Der Einstieg in ein Projekt kann oft unübersichtlich sein – umso wichtiger ist es, den Projektauftrag gründlich zu verstehen. Aus diesem Grund haben wir uns intensiv mit dem bereitgestellten Projektdokument (siehe Anhang) auseinandergesetzt. Dieses enthält zentrale Vorgaben sowie Anforderungen an die geplante Webapplikation.

Beim Lesen sind uns verschiedene Fragen die schon vorgegeben Fragen ins Auge gekommen, die wir im Anschluss gemeinsam besprochen haben. Durch diesen Austausch konnten wir ein klares Bild der Projektziele entwickeln und sicherstellen, dass alle Beteiligten die gleichen Erwartungen und ein gemeinsames Verständnis der Aufgabe haben.

2.2 Planen

2.2.1 Aufgabenplanung

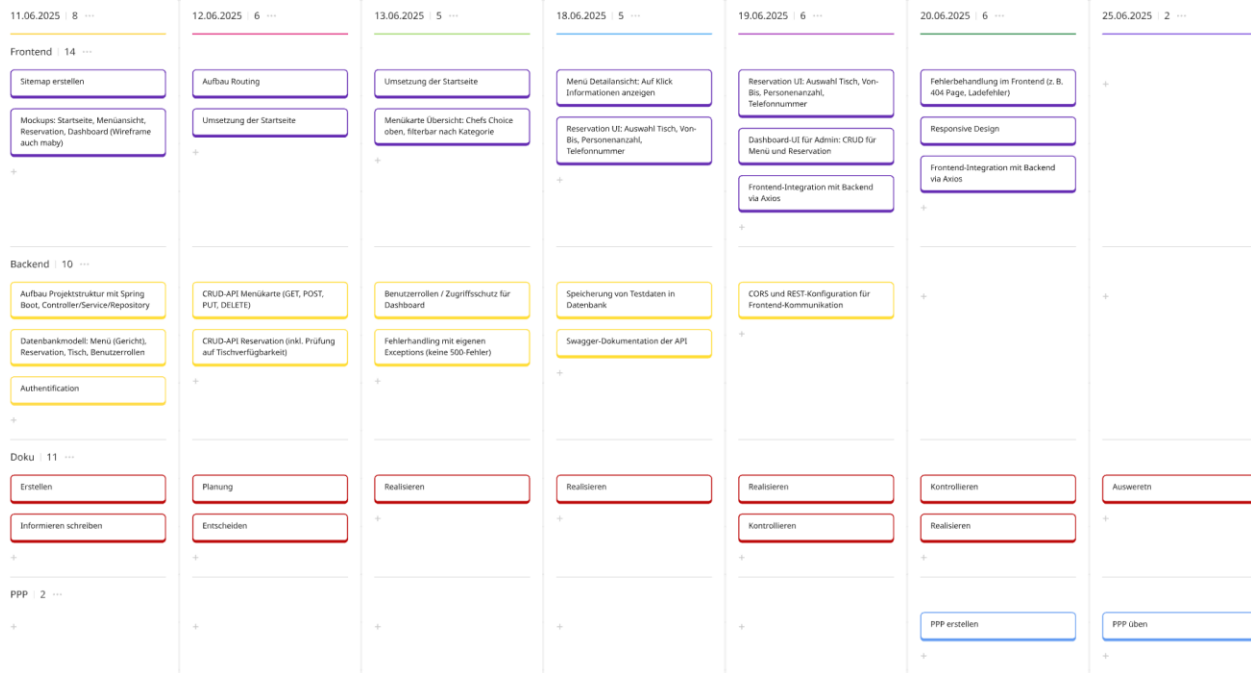
Zu Beginn der Planen-Phase stellten wir uns folgende Frage: Wie können wir unsere Aufgaben übersichtlich zeitlich und inhaltlich darstellen? Hierfür eignete sich für uns ein Kanban, welches auf unserem Miro-Board platziert wurde. Den Vorgaben folgend sollte jeder einen Teil zur Dokumentation sowie zur Entwicklung des Codes beitragen. Zur Übersicht sind unsere Designs farbig nach Frontend und Backend aufgeteilt.



2.2.2 Ablaufplanung

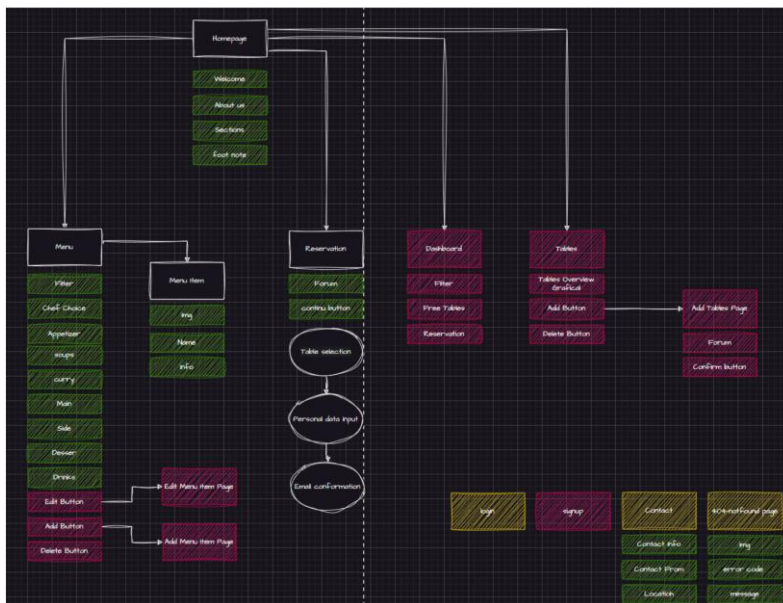
Um den zeitlichen Ablauf der nächsten Tage zu planen, half uns eine Roadmap. Diese zeigte unsere Aufgaben aufgeteilt in Frontend Backend Präsentation und Dokumentation. Dabei wird ersichtlich, wann eine Aufgabe abgeschlossen werden sollte und Meilensteine können besser nachvollzogen werden.

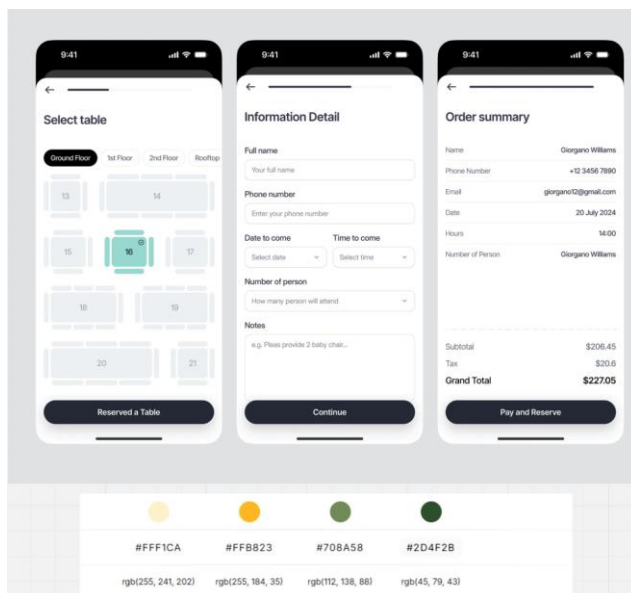
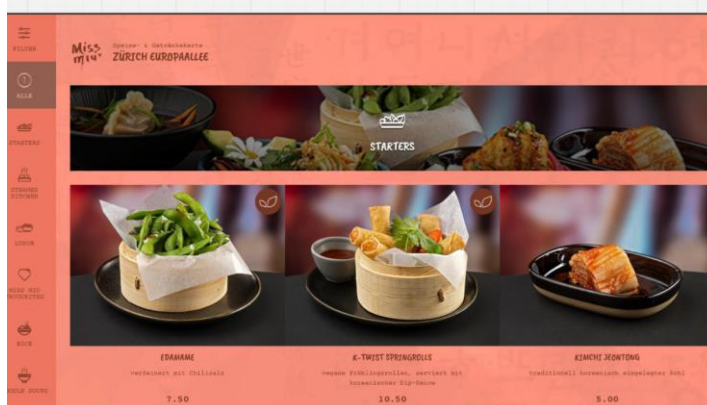
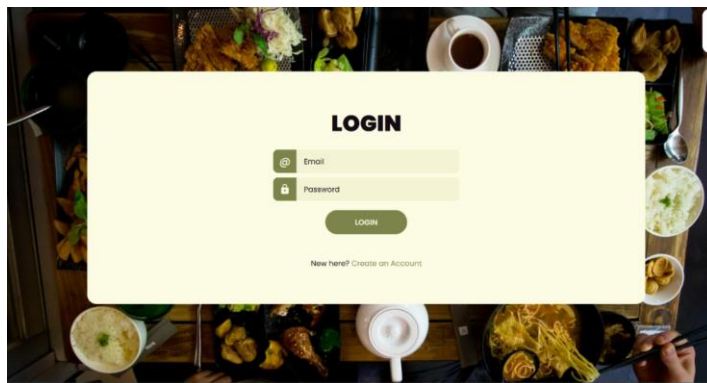
Fullstack Applikation



2.2.3 Frontendplanung

Parallel zur zeitlichen Planung erarbeiteten wir auch den **inhaltlichen Aufbau der Website**. Wir definierten die wichtigsten Seiten (z. B. Startseite, Reservierung, Speisekarte, Kontakt). Dabei legten wir besonderen Wert auf Benutzerfreundlichkeit und eine klare Navigation.





2.2.4 Programmstruktur und Aufgabenverteilung

Zur weiteren Umsetzung unseres Projekts planen wir den Aufbau des Codes mithilfe eines Ablaufdiagramms in draw.io. Darin hielten wir fest, welche Logik in Node-RED umgesetzt wird und wie die einzelnen Komponenten miteinander kommunizieren. Ein Activity Diagramm half dabei, den Datenfluss zwischen den Systemteilen sowie deren Verbindung zu Node-RED visuell darzustellen.

2.3 Entscheiden

Für die Umsetzung teilten wir die Aufgaben klar im Team auf: Nepomuk und Diego übernahmen das Backend, während Sylia und Luka sich um das Frontend kümmerten. Diego richtete die Datenbank ein und implementierte die benötigten Verbindungen, während Nepomuk die Backend-Logik entwickelte und die Struktur vorgab.

Gemeinsam entschieden wir uns für eine thailändische Menükarte als Inhalt. Die Daten dazu wurden grösstenteils von Luka und Sylia verfasst, teilweise auch durch KI-Unterstützung ergänzt. Diego übernahm anschliessend die Integration dieser Inhalte in die Datenbank.

2.4 Realisieren

2.4.1 Frontend

Für die Umsetzung des Frontends orientierten wir uns an den vorab definierten Anforderungen und erstellten darauf basierend die notwendigen Benutzeroberflächen. Wir verwendeten React als Framework, um eine modulare und wiederverwendbare Struktur zu gewährleisten. Dabei implementierten wir verschiedene Komponenten, wie beispielsweise Formulare zur Reservierung, Menüanzeigen und Dropdowns zur Auswahl der Anzahl Personen. Besonderen Wert legten wir auf eine intuitive Nutzerführung und eine responsive Gestaltung, damit die Anwendung auf unterschiedlichen Geräten einwandfrei funktioniert.

Die Anbindung an das Backend erfolgte über REST-API-Aufrufe, mit denen wir Daten abrufen und übermitteln konnten. Zur Validierung der Benutzereingaben setzten wir Bibliotheken wie Formik ein, um eine direkte Rückmeldung bei fehlerhaften Eingaben zu gewährleisten. Während der Entwicklung testeten wir die Frontend-Komponenten kontinuierlich, um eine hohe Benutzerfreundlichkeit und Stabilität sicherzustellen.

2.4.2 Backend

Das Backend bildet die zentrale Schicht zur Verwaltung der Geschäftsdaten. Es implementiert die CRUD-Methoden für alle Kernfunktionen der App, darunter Menüverwaltung, Reservierungserstellung sowie Abfrage der Tischverfügbarkeit. Zur Speicherung der Daten wurde eine relationale Datenbank aufgebaut, die alle relevanten Informationen in Tabellen organisiert und mit dem Backend verbunden ist.

Zur Sicherstellung der Funktionalität und Stabilität testeten wir die Endpunkte intensiv mit Postman. Dies ermöglichte es uns, die korrekte Verarbeitung von Anfragen sowie die Einhaltung der Geschäftslogik zu überprüfen. Nach erfolgreicher Entwicklung wurde das Backend nahtlos mit dem Frontend verbunden, sodass eine durchgängige und dynamische Nutzererfahrung gewährleistet ist.

2.5 Kontrollieren

Nach der erfolgreichen Umsetzung der Webapplikation begannen wir mit der **Kontrollieren-Phase**, um sicherzustellen, dass die Anwendung korrekt funktioniert und den definierten Anforderungen entspricht. Unser Hauptziel war es, die Funktionalität, Stabilität und Benutzerfreundlichkeit zu prüfen und gegebenenfalls nachzubessern.

2.5.1 Funktionsprüfung

Zunächst testeten wir die zentralen Funktionen der Webapplikation:

- Reservierungen: Wir überprüften, ob neue Reservierungen korrekt gespeichert und im System angezeigt werden.
- Menüanzeige: Das Menü wurde auf Vollständigkeit und korrekte Darstellung getestet.
- Buttons: Wir stellten sicher, dass alle Schaltflächen die vorgesehenen Aktionen ausführen, z. B. Weiterleitungen oder Datenübermittlungen.

Auch die Verbindung zwischen Frontend und Backend wurde genau kontrolliert. Dabei achteten wir insbesondere auf Datenflüsse, Ladezeiten und korrekte Rückmeldungen bei Benutzeraktionen.

2.5.2 API-Tests mit Postman

Zur gezielten Überprüfung der Schnittstellen verwendeten wir Postman. Damit testeten wir systematisch alle relevanten API-Routen:

- GET-, POST-, PUT- und DELETE-Requests wurden mit verschiedenen Parametern getestet.
- Die Rückmeldungen der Datenbank wurden auf Konsistenz und Richtigkeit geprüft.
- Fehlerfälle (z. B. ungültige Eingaben oder leere Felder) wurden simuliert, um das Verhalten der Anwendung zu beobachten.

Durch diese Tests konnten wir sicherstellen, dass die Applikation auch im Backend korrekt auf Anfragen reagiert.

2.5.3 Fehlererkennung und Behebung

Während des Testens entdeckten wir kleinere Fehler und Schwächen, z. B.:

- Falsche Eingaben, die nicht korrekt abgefangen wurden.
- Darstellungsprobleme im User Interface, etwa bei bestimmten Bildschirmgrößen oder Eingabekombinationen.

Diese Fehler wurden direkt identifiziert, dokumentiert und umgehend behoben. Dabei führten wir jeweils Regressionstests durch, um sicherzustellen, dass keine neuen Probleme durch die Korrekturen entstanden.

2.5.4 Stabilitäts- und Usability-Verbesserung

Durch wiederholte Testzyklen konnten wir die Stabilität der Webapplikation deutlich erhöhen. Ebenso flossen Erkenntnisse zur Benutzerfreundlichkeit direkt in kleinere Anpassungen am UI/UX-Design ein – z. B. durch klarere Rückmeldungen bei Aktionen oder angepasste Buttonplatzierungen.

2.5.5 Fazit der Kontrollieren-Phase

Die umfangreichen Tests in dieser Phase bestätigten die Funktionsfähigkeit und Qualität der Applikation. Alle Kernanforderungen wurden erfüllt. Kleinere Schwächen konnten behoben werden, bevor die Applikation produktiv eingesetzt wurde. Damit war die Kontrollieren-Phase erfolgreich abgeschlossen und der Weg frei für die nächste Etappe im Projekt.

2.6 Auswerten

Das Projekt konnte nicht vollständig erfolgreich abgeschlossen werden. Die Umsetzung der Menüseite (Priorität 1) wurde realisiert, allerdings ohne die ursprünglich geplante Filterfunktion. Die Reservation (Priorität 2) funktioniert grundsätzlich, weist jedoch noch viele Bugs auf und ist in der aktuellen Form nicht vollständig benutzerfreundlich. Die Funktionalitäten mit Priorität 3 konnten aus zeitlichen Gründen nicht mehr umgesetzt werden.

Im Backend wurden zwar sämtliche Funktionen implementiert, allerdings erfolgten nur unzureichende Nachfragen oder Abstimmungen mit dem Frontend-Team. Dadurch entstanden zahlreiche Unklarheiten und Probleme beim Zusammenspiel der beiden Systeme. Dies führte dazu, dass im Frontend ein erheblicher zeitlicher Aufwand für Bugfixes und Anpassungen betrieben werden musste, um die Schnittstellenfehler zu kompensieren. Insgesamt war die zur Verfügung stehende Zeit für den Umfang des Projektes zu knapp bemessen. Zusätzlich gestaltete sich die Kommunikation im Team allgemein schwierig und kompliziert, was den Projektfortschritt weiter beeinträchtigte.

2.6.1 Reflexion

Nepomuk:

Projektleitung und Teamarbeit

Als Projektleiter musste ich dafür sorgen, dass alle im Team gut zusammenarbeiten. Das hat funktioniert, weil mein Team motiviert war und alle mitgemacht haben.

Tägliche Besprechungen

Ich habe jeden Tag kurze Meetings gemacht, damit alle wissen, was läuft. Was mir aufgefallen ist: Die Meetings wurden immer kürzer. Nach ein paar Tagen wussten wir, wer was gut kann, und konnten die Aufgaben besser verteilen.

Backend-Arbeit

Diego und ich haben das Backend zusammen gemacht. Da wir schon mal zusammengearbeitet hatten, wussten wir von Anfang an, wer welche Stärken hat. Das hat die Planung viel einfacher gemacht.

Diego: Während dieses Projekts hatte ich nur wenige Schwierigkeiten. Ich verstand das Backend sehr gut, was mir die Arbeit deutlich erleichterte. Auch der Umgang mit der SQL-Datenbank fiel mir leicht, da ich bereits Erfahrung damit hatte. Beim Arbeiten zu Hause profitierte ich von der ruhigen Umgebung und der ausreichenden Zeit, um mich auf meine Aufgaben zu konzentrieren. Dadurch konnte ich effizient und ohne grossen Zeitdruck am Projekt arbeiten. In der Noser Young funktionierte die Zusammenarbeit ebenfalls sehr gut. Die Aufgaben waren sinnvoll aufgeteilt, was zu einem reibungslosen Ablauf beitrug.

Sylia: Das Projekt war insgesamt in Ordnung, allerdings bevorzuge ich persönlich eher Einzelarbeit. Die Umsetzung der Reservierungsfunktion hatte viele Bugs und Herausforderungen. Die Zeit war insgesamt knapp, was zusätzlichen Druck erzeugt hat. Die Kommunikation und Mitarbeit mit dem Backend-Team war schwierig. Absprachen wurden oft nicht eingehalten, was dazu geführt hat, dass viele Fehler im Frontend erst durch fehlende oder fehlerhafte Backend-Daten entstanden sind. Ein grosser Teil der vorgesehenen Entwicklungszeit musste daher in die Behebung von Frontend-Bugs investiert werden, die eigentlich Backend-seitig hätten verhindert werden können. Die Grundstruktur des Projekts war nicht optimal geplant, was immer wieder zu Verzögerungen und unnötigem Mehraufwand geführt hat. Besonders die Reservierungslogik war problematisch und fehleranfällig. Trotz der Schwierigkeiten konnte ich viel über Fehlerbehandlung, Debugging und pragmatische Lösungsansätze lernen. Für zukünftige Projekte wären eine klarere Aufgabenverteilung, bessere Backend-API-Planung und verbindlichere Teamkommunikation sehr wichtig.

3 Schluss

Zum Abschluss unseres Projekts „RestFoods“ können wir festhalten, dass wir ein funktionierendes und benutzerfreundliches Fullstack-System erfolgreich umgesetzt haben. Durch die Arbeit an Frontend, Backend und Datenbankbindung konnten wir ein tiefes Verständnis für den Aufbau und die Funktionsweise moderner Webapplikationen gewinnen.

