



Entwicklung eines Webshops

Studienarbeit

an der

DHBW Stuttgart

von

Masoud Abdulhanan (9769728)

16.06.2024

Abstract

Inhaltsverzeichnis

1 Einleitung	1
1.1 Hintergrund und Motivation	1
1.2 Zielsetzung der Studienarbeit	2
1.3 Relevanz des Themas im Kontext des E-Commerce	3
2 Analysephase	6
2.1 Definition der Anforderungen	6
2.2 Zielgruppenanalyse	6
2.3 Recherche und Analyse bestehender Webshops und E-Commerce-Strategien	7
3 Technologieauswahl	8
3.1 Bun	8
3.1.1 Funktionen von Bun	8
3.1.2 Bun vs. Node.js	9
3.2 Hono	10
3.2.1 Vorteile von Hono	10
3.2.2 Hono vs. Express.js	11
Quellverzeichnis	13

1 Einleitung

1.1 Hintergrund und Motivation

Die digitale Transformation hat in den letzten Jahrzehnten das Gesicht des Einzelhandels radikal verändert. Die Entwicklung des E-Commerce hat nicht nur neue Geschäftsmodelle ermöglicht, sondern auch die Erwartungen und das Verhalten der Verbraucher grundlegend verändert. Diese Veränderung wird durch die weitreichende Verfügbarkeit des Internets und die zunehmende Nutzung mobiler Geräte weiter beschleunigt. Studien zeigen, dass der E-Commerce-Markt weltweit boomt, was auf die zahlreichen Vorteile zurückzuführen ist, die diese Einkaufsform sowohl für Unternehmen als auch für Verbraucher bietet [1], [2].

E-Commerce, oder elektronischer Handel, bezeichnet den Kauf und Verkauf von Waren und Dienstleistungen über das Internet. Dabei werden verschiedene Zahlungsmethoden wie Kreditkarten, Debitkarten, Online-Banking und Zahlung über digitale Geldbörsen genutzt [2]. Die Verfügbarkeit einer breiten Produktpalette zu wettbewerbsfähigen Preisen und die Möglichkeit, bequem von zu Hause aus zu shoppen, haben den E-Commerce zu einer bevorzugten Einkaufsform gemacht. Die bedeutendsten Treiber dieses Wachstums sind die zunehmende Verbreitung des Internets, die verbesserte Infrastruktur für digitale Zahlungen und der Einsatz von Technologien wie Big Data und Künstliche Intelligenz (KI), die personalisierte Einkaufserlebnisse ermöglichen [3].

Ein besonderer Aspekt des E-Commerce, der in den letzten Jahren erheblich an Bedeutung gewonnen hat, ist das Dropshipping. Dropshipping ist ein Geschäftsmodell, bei dem Einzelhändler keine eigenen Lagerbestände halten, sondern Bestellungen direkt vom Großhändler oder Hersteller an den Endkunden versenden lassen. Dies reduziert die Anfangsinvestitionen und das finanzielle Risiko für neue Unternehmen erheblich und ermöglicht es ihnen, eine breite Produktpalette anzubieten, ohne große Lagerbestände verwalten zu müssen [2].

Die wachsende Popularität von E-Commerce und Dropshipping wird durch beeindruckende Marktstatistiken unterstützt. Der globale E-Commerce-Markt verzeichnete 2020 ein Umsatzvolumen von 4,28 Billionen US-Dollar und wird voraussichtlich bis 2023 auf 6,54 Billionen US-Dollar anwachsen. Dies entspricht einer durchschnittlichen jährlichen Wachstumsrate (CAGR) von etwa 10,7% [1].

Die Abbildung unten zeigt die meistbesuchten Websites in Deutschland im März 2024. Dabei ist deutlich zu erkennen, dass große E-Commerce-Plattformen wie Amazon eine signifikante Anzahl monatlicher Besuche verzeichnen, was die Relevanz und den Einfluss von E-Commerce im digitalen Raum unterstreicht

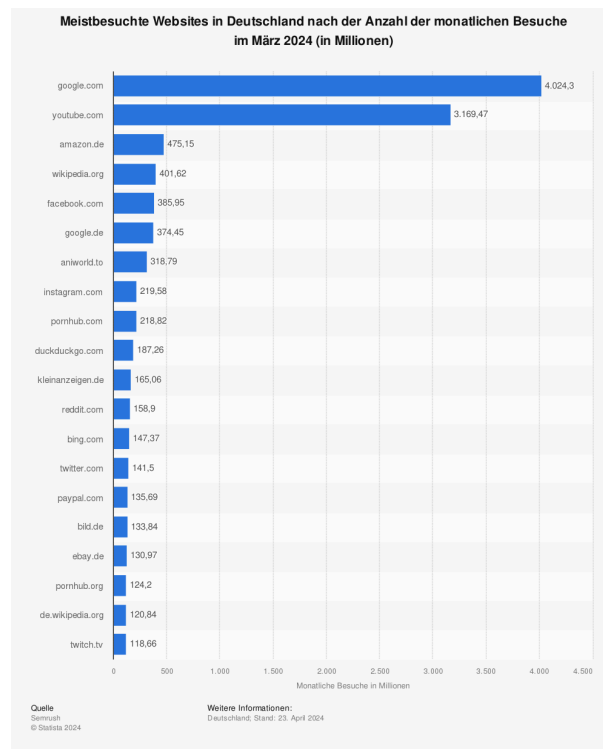


Abbildung 1: Meistbesuchte Webseiten in Deutschland im März 2024, Quelle: Statista

1.2 Zielsetzung der Studienarbeit

Die vorliegende Studienarbeit zielt darauf ab, einen modernen Webshop zu konzipieren und umzusetzen, der den aktuellen Anforderungen des E-Commerce entspricht. Dabei sollen sowohl funktionale Anforderungen wie Benutzerkontenverwaltung, Artikelanzeige und -suche, Warenkorb und Kassenfunktionen als auch nicht-funktionale Anforderungen wie Performance, Skalierbarkeit und Sicherheit berücksichtigt werden [3]. Ein besonderer Fokus liegt auf der Integration eines Adminbereichs, der die effiziente Verwaltung des Webshops ermöglicht.

Die Arbeit umfasst die Analyse bestehender E-Commerce-Plattformen, um Best Practices zu identifizieren und innovative Ansätze zu entwickeln, die den Webshop sowohl benut-

zerfreundlich als auch technisch robust machen. Methodisch wird die Arbeit durch eine Kombination aus Literaturrecherche, Analyse bestehender Webshops und der praktischen Umsetzung eines eigenen Prototyps durchgeführt [3].

1.3 Relevanz des Themas im Kontext des E-Commerce

Im Kontext des E-Commerce ist die Entwicklung eines effizienten und benutzerfreundlichen Webshops von zentraler Bedeutung. Der Erfolg eines Online-Shops hängt maßgeblich von der Benutzererfahrung ab, die durch ein intuitives Design, schnelle Ladezeiten und eine sichere Zahlungsabwicklung gewährleistet wird. Ein gut gestalteter Webshop kann die Kundenzufriedenheit erhöhen, die Kundenbindung stärken und somit den Umsatz steigern [1], [2].

Darüber hinaus spielt der E-Commerce eine wichtige Rolle in der heutigen Wirtschaft, indem er Unternehmen ermöglicht, ihre Reichweite zu erweitern und neue Märkte zu erschließen. Insbesondere das Dropshipping-Modell bietet eine attraktive Möglichkeit für Start-ups und kleine Unternehmen, ohne große Anfangsinvestitionen in den Markt einzutreten und von den Vorteilen des Online-Handels zu profitieren [2].

Die folgende Abbildung zeigt die prognostizierte Anzahl der Social-Commerce-Nutzer in den USA bis 2023, was die zunehmende Bedeutung und das Wachstumspotenzial des E-Commerce im sozialen Kontext verdeutlicht

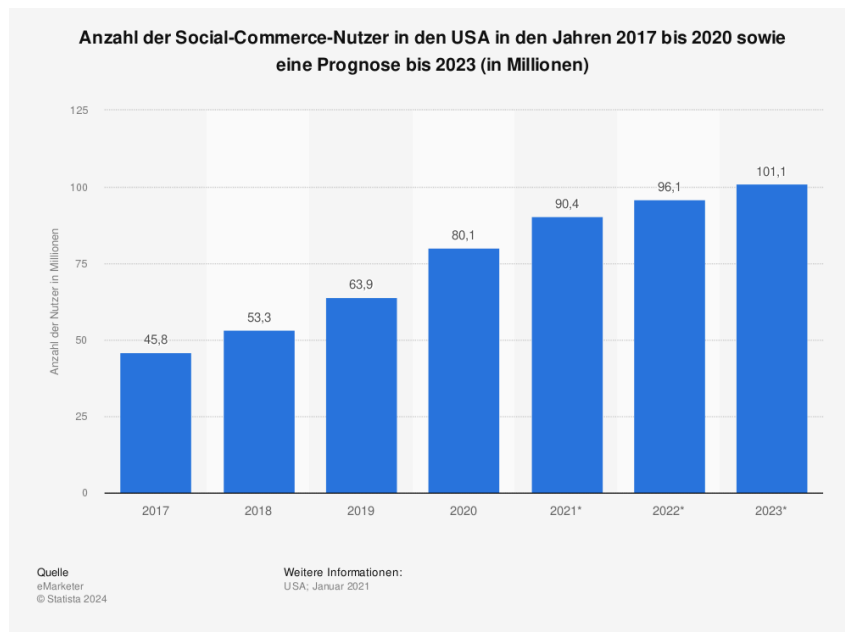


Abbildung 2: Anzahl der Social-Commerce-Nutzer in den USA

Zusätzlich zeigt die Abbildung die Anzahl der Transaktionen über PayPal weltweit, was die wachsende Nutzung digitaler Zahlungssysteme und die damit verbundene Sicherheit und Effizienz im E-Commerce unterstreicht

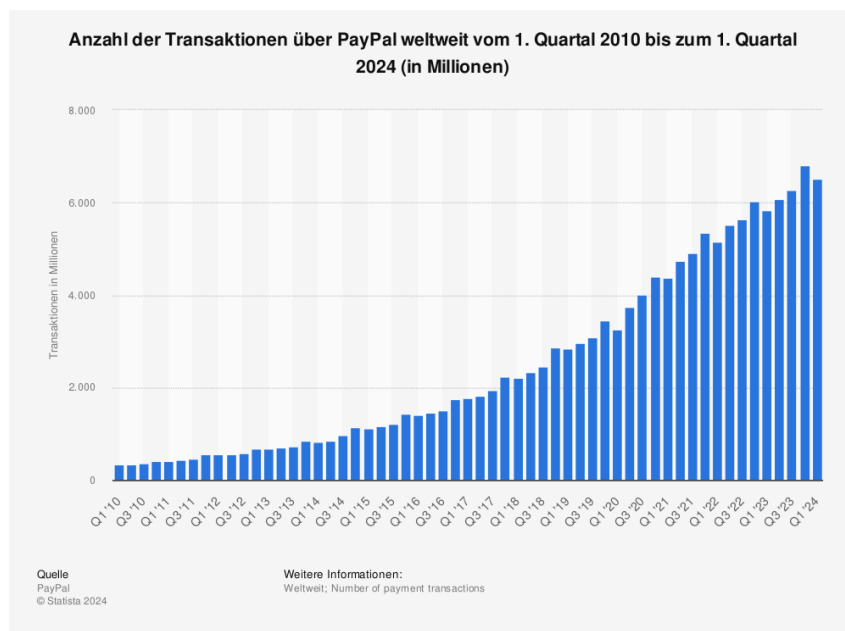
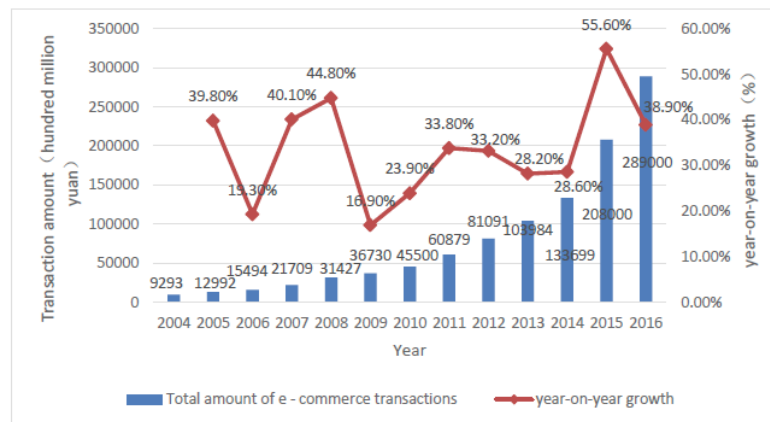


Abbildung 3: Anzahl der Transaktionen über PayPal weltweit

Die Abbildungen unten zeigen das Wachstum des Online-Einzelhandelsmarktes in China sowie das Gesamtvolumen der E-Commerce-Transaktionen von 2004 bis 2016. Diese Dia-

gramme illustrieren die rasante Entwicklung und die wirtschaftliche Bedeutung des E-Commerce in einem der größten Märkte weltweit

Figure 1 China's total E-commerce transactions from 2004 to 2016



Source: China's E-commerce report, 2015 and 2016 data from Ali Research Institute and Avatar large data processing center.

Abbildung 4: Wachstum des Online-Einzelhandelsmarktes in China

Figure 2 China Online retail trade volume from 2008 to 2016



Source: China's E-commerce report. 2016 data provided by Ali Research Institute and consolidated by the research group.

Abbildung 5: Gesamtvolumen der E-Commerce-Transaktionen in China

2 Analysephase

2.1 Definition der Anforderungen

Die Definition der Anforderungen an den Webshop umfasst sowohl funktionale als auch nicht-funktionale Aspekte. Zu den funktionalen Anforderungen zählen die Verwaltung von Benutzerkonten, die Bereitstellung einer benutzerfreundlichen Such- und Filterfunktion, die Integration eines Warenkorbsystems sowie die sichere Abwicklung von Zahlungen [3].

Nicht-funktionale Anforderungen betreffen die Performance, Skalierbarkeit und Sicherheit des Webshops. Hierbei ist sicherzustellen, dass der Webshop auch bei hohen Zugriffszahlen stabil und zuverlässig funktioniert. Zudem müssen Datenschutzrichtlinien eingehalten werden, um die Sicherheit der Kundendaten zu gewährleisten.

2.2 Zielgruppenanalyse

Die Zielgruppenanalyse ist ein entscheidender Schritt in der Entwicklung eines Webshops. Sie dient dazu, die Bedürfnisse und Erwartungen der potenziellen Nutzer zu verstehen und den Webshop entsprechend auszurichten. Dabei werden demografische Daten, Einkaufsverhalten und Präferenzen der Zielgruppe untersucht.

Eine erfolgreiche Zielgruppenanalyse ermöglicht es, den Webshop so zu gestalten, dass er die gewünschten Kundensegmente anspricht und eine hohe Benutzerzufriedenheit erzielt. Dies beinhaltet auch die Anpassung des Designs und der Funktionalitäten an die spezifischen Anforderungen der Zielgruppe [3].

2.3 Recherche und Analyse bestehender Webshops und E-Commerce-Strategien

Die Recherche und Analyse bestehender Webshops bietet wertvolle Einblicke in bewährte E-Commerce-Strategien und Designprinzipien. Durch die Untersuchung von erfolgreichen Online-Shops können Best Practices identifiziert und Schwachstellen vermieden werden.

Zu den analysierten Aspekten gehören die Benutzerfreundlichkeit, das Design, die Navigationsstruktur und die Integration von Zahlungs- und Versandoptionen. Darüber hinaus werden auch Marketingstrategien und Kundenbindungsmaßnahmen bestehender Webshops untersucht, um deren Effektivität zu bewerten und gegebenenfalls zu adaptieren [3].

Durch die Kombination aus theoretischem Wissen und praktischen Einblicken wird eine solide Grundlage für die Entwicklung eines leistungsfähigen und benutzerfreundlichen Webshops geschaffen.

3 Technologieauswahl

In diesem Kapitel wird der Entwurf des Systems beschrieben. Ziel des Kapitels ist es, einen detaillierten Überblick über die Systemarchitektur und die Technologieauswahl zu geben sowie die einzelnen Komponenten des Systems darzustellen. Es wird erläutert, warum bestimmte Technologien und Architekturen gewählt wurden und wie die verschiedenen Komponenten des Systems miteinander interagieren.

Zunächst wird die Auswahl der verwendeten Technologien und Frameworks begründet. Anschließend wird die Systemarchitektur detailliert beschrieben, gefolgt vom Datenbankentwurf und dem API-Design. Ein weiterer Abschnitt widmet sich der Implementierung der Authentifizierung und Benutzerverwaltung, gefolgt von der Datenvalidierung. Die Backend-Logik wird ebenfalls im Detail erläutert.

3.1 Bun

Bun ist eine JavaScript-Runtime Umgebung für den Server, die anders als Node.js oder Deno nicht auf der V8-Engine basiert, sondern auf einer eigenen JavaScript-Engine, welche mithilfe von Apples WebKit Engine implementiert wurde. Bun wurde zudem in einer „low-level general“ Programmiersprache namens Zig geschrieben, welche von Rust und C inspiriert ist. Die Entscheidung für Bun fiel aufgrund der hohen Performance und Sicherheit, die durch die Verwendung von Zig und der WebKit-Engine gewährleistet wird. Bun ermöglicht es, serverseitige Anwendungen in JavaScript zu entwickeln und auszuführen. Bun wurde von Jarred Summer entwickelt und ist eine Open-Source-Software, die unter der MIT-Lizenz veröffentlicht wird. Die erste offizielle Version von Bun (Bun 1.0) wurde im September 2023 veröffentlicht.

3.1.1 Funktionen von Bun

Bun bietet eine Reihe von Funktionen, die es zu einer Plattform für die Entwicklung von serverseitigen Anwendungen machen. Dazu gehören:

- Kompatibilität mit Node.js
- Hohe Laufleistung und geringer Speicherverbrauch

- Vereinfachte Modulverwaltung
- TypeScript-Unterstützung
- Web-Standard-APIs
- JSX-Unterstützung
- Watch-Modus für automatisches Neuladen von Änderungen
- Cross-Plattform-Unterstützung

3.1.2 Bun vs. Node.js

Anders als Node.js ist bun nicht auf npm angewiesen und benötigt keine externen Abhängigkeiten zur Ausführung. Stattdessen wird eine integrierte Standardbibliothek verwendet, die Funktionen wie HTTP-Server, Dateisystemzugriff und Netzwerkkommunikation bereitstellt. Dies macht die Entwicklung und Bereitstellung von Anwendungen mit Bun einfacher und sicherer. Bun basiert zudem anders als Node.js nicht auf der von Google entwickelten V8-Engine, sondern auf einer Erweiterung von JavaScriptCore, die von Apple entwickelt und bereitgestellt wird. JSC priorisiert schnellere Startzeiten und geringeren Speicherverbrauch, was zu einer etwas langsameren Ausführungsgeschwindigkeit führt. V8 priorisiert hingegen die Ausführungsgeschwindigkeit mit mehr Runtime-Optimierungen, was zu einem höheren Speicherverbrauch führen kann. Das führt dazu, dass Bun bis zu 4xmal so schnell startet als Node.js

```
~/Desktop
> hyperfine "bun hello.js" "node hello.js" "deno run hello.js" --warmup=100
Benchmark 1: bun hello.js
Time (mean ± σ):    5.2 ms ± 0.4 ms    [User: 3.1 ms, System: 1.4 ms]
Range (min ... max): 4.9 ms ... 6.7 ms    345 runs

Warning: Command took less than 5 ms to complete. Results might be inaccurate.

Benchmark 2: node hello.js
Time (mean ± σ):    25.1 ms ± 0.6 ms    [User: 21.6 ms, System: 2.7 ms]
Range (min ... max): 24.4 ms ... 27.7 ms    105 runs

Benchmark 3: deno run hello.js
Time (mean ± σ):    11.4 ms ± 0.6 ms    [User: 8.9 ms, System: 2.1 ms]
Range (min ... max): 10.8 ms ... 14.1 ms    202 runs

Summary
'bun hello.js' ran
  2.19 ± 0.19 times faster than 'deno run hello.js'
  4.81 ± 0.34 times faster than 'node hello.js'

~/Desktop took 13s
> cat hello.js
File: hello.js
1 console.log("hi");
```

Abbildung 6: Bun vs. Node.js - Startzeitvergleich, Quelle: Builder.io

Die Benchmark-Ergebnisse, welche in Abbildung 6 gezeigt werden, zeigen eine Verbesserung von mehr als siebzehnmal so schnell wie übliche Paketmanager.

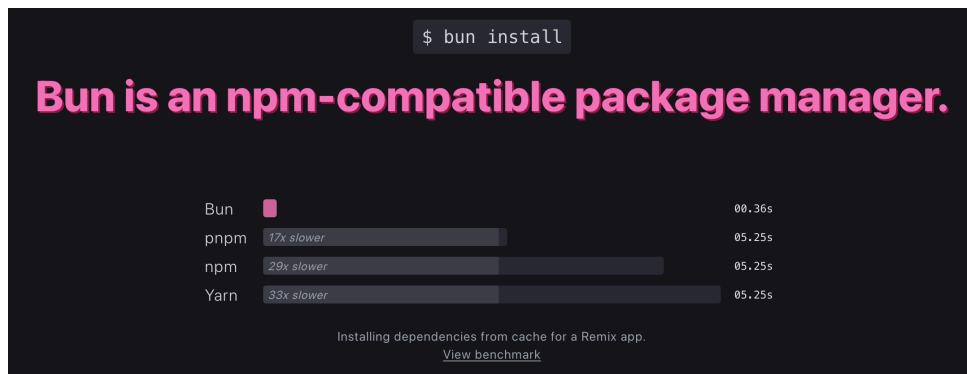


Abbildung 7: Bun vs. andere Paketmanager - Benchmark-Ergebnisse, Quelle: Bun

Während Node.js eine gute Runtime-Umgebung für JavaScript ist, werden TypeScript Dateien in Node.js nicht direkt unterstützt. TypeScript-Dateien müssen zuerst in JavaScript-Dateien kompiliert werden, bevor sie in Node.js ausgeführt werden können. Bun hingegen unterstützt TypeScript-Dateien direkt, was die Entwicklung von serverseitigen Anwendungen in TypeScript vereinfacht. TypeScript-Dateien können direkt mit dem Befehl `bun „Dateiname.ts“` ausgeführt werden.

Deshalb wurde Bun als Server-Runtime für die Entwicklung des Webshops gewählt, da es eine hohe Performance, Sicherheit und TypeScript-Unterstützung bietet und die Entwicklung von serverseitigen Anwendungen in JavaScript und TypeScript vereinfacht.

3.2 Hono

Hono ist einfaches und ultraschnelles Web-Framework, welches auf jeder JavaScript-Run-time-Umgebung läuft. Entwickelt wurde Hono von Yusuke Wada und ist eine Open-Source-Software, die unter der MIT-Lizenz veröffentlicht wird. Hono wurde speziell für die Entwicklung von Webanwendungen und APIs entwickelt und bietet eine Reihe von Funktionen, die es zu einer leistungsstarken Plattform für die Entwicklung von Webanwendungen machen.

3.2.1 Vorteile von Hono

Die Entscheidung für Hono als Web-Framework wurde aufgrund mehrerer Schlüsselfaktoren getroffen:

- **Ultraschnell und effizient:** Der Router „RegExpRouter“ ist besonders schnell und arbeitet nicht mit linearen Schleifen, was eine schnelle und effiziente Routenauflösung

ermöglicht. Dies macht Hono ideal für Anwendungen, die eine hohe Geschwindigkeit und geringe Latenz erfordern.

- **Leichtgewichtig und modular:** Hono ist äußerst leichtgewichtig und hat keine externen Abhängigkeiten. Mit dem hono/tiny-Preset beträgt die Größe von Hono weniger als 14 kB, was im Vergleich zu anderen Frameworks sehr kompakt ist. Trotz seiner geringen Größe bietet Hono eine Vielzahl von Middleware- und Hilfsfunktionen, die es einfach machen, leistungsstarke Anwendungen zu entwickeln.
- **Multi-Runtime-Unterstützung:** Hono ist äußerst vielseitig und kann auf einer Vielzahl von Plattformen eingesetzt werden, darunter Cloudflare Workers, Fastly Compute, Deno, Bun, AWS Lambda und Node.js. Dadurch ist es möglich, die gleiche Codebasis auf verschiedenen Plattformen zu verwenden, was die Entwicklung und Wartung von Anwendungen vereinfacht.
- **Middleware inklusive:** Hono bietet eine umfangreiche Sammlung von Middleware, benutzerdefinierten Middleware und Hilfsfunktionen, die es Entwicklern ermöglichen, weniger Code zu schreiben und mehr zu erreichen. Von der Basic Authentication bis zur GraphQL Server-Unterstützung bietet Hono alles, was für die Entwicklung leistungsstarker Webanwendungen erforderlich ist. Zudem auch einen kleinen Validator für die Datenvalidierung, um die Datenintegrität zu gewährleisten.

3.2.2 Hono vs. Express.js

Im Vergleich zu Express.js, einem der beliebtesten Web-Frameworks für Node.js, bietet Hono eine Reihe von Vorteilen:

Vorteile von Hono:

- **Mikroservices-Architektur:** Hono ist speziell für Mikroservices-Architekturen ausgelegt, was die Skalierbarkeit und Modularität von Anwendungen erleichtert.
- **Leistung und Skalierbarkeit:** Hono bietet Leistungsbenchmarks und effiziente Anfrageverarbeitung, was besonders für hochskalierbare Anwendungen von Vorteil ist.
- **Eingebaute WebSocket-Unterstützung:** Hono bietet WebSocket-Unterstützung für die Implementierung von Echtzeitfunktionen an.
- **TypeScript-Unterstützung:** Hono unterstützt TypeScript nativ, was für Typsicherheit und verbesserte Entwicklerwerkzeuge sorgt.

- Aktive Community-Wartung: Hono wird von einer aktiven Entwicklergemeinschaft gepflegt, was regelmäßige Updates und Verbesserungen gewährleistet.

Deshalb wurde Hono als Web-Framework für die Entwicklung des Webshops gewählt, da es vor allem mit Kombination von Bun ultraschnell, effizient, leichtgewichtig, modular und vielseitig ist.

Framework	Runtime	Durchschnitt	Ping	Query	Body
Hono	bun	184,966.48	234,593.57	185,108.2	135,197.67
Hono	Node	42,699.317	60,797.19	56,645.8	10,654.96
Express	node	16,461.68	17,656.74	16,615.32	15,112.98

Die Ergebnisse sind in req/s gemessen

Quellverzeichnis

- [1] Yue Hongfei, Wu Jiani, und L. Yao, „Analysis on the Development and Bottlenecks of China's Cross - Border E – Commerce“. 2017. Zugegriffen: 26. Mai 2024. [Online]. Verfügbar unter: https://www.unido.org/sites/default/files/2017-10/WP_17_2017.pdf
- [2] Gurpreet Singh, Harjot Kaur, und Amitpal Singh, „Dropshipping in E-Commerce - A Perspective“. 2018. Zugegriffen: 26. Mai 2024. [Online]. Verfügbar unter: <https://dl.acm.org/doi/10.1145/3271972.3271993>
- [3] Lena-Maria Mikat und Prof Dr. Monika Engelen, „Online-Marketing Basics für B2B Praktische Handlungsempfehlungen und Best Practices“. 2018. Zugegriffen: 26. Mai 2024. [Online]. Verfügbar unter: https://epb.bibl.th-koeln.de/frontdoor/deliver/index/docId/1152/file/PraxisPaper_Online+Marketing+f%C3%BCr+B2B.pdf