

# LLM-gestuetzte-Softwareentwicklung-Labor

aus dem Studiengang LLM-gestuetzte-Softwareentwicklung

an der Hochschule Esslingen University of Applied Sciences Campus  
Esslingen Flandernstraße

von

Tim Jauch

Annabel Heberle

22.10.2025

Matrikelnummer, Kurs: 763086, SWB  
770677, SWB

Betreuer an der HSE: Prof. Dr. Jörg Nitzsche, Dr. Stefan Kaufmann, Klemens Morbe

# Selbstständigkeitserklärung

Wir versichern hiermit, dass wir unsere Arbeit mit dem Thema:

## **LLM-gestuetzte-Softwareentwicklung-Labor**

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt haben. Wir versichern zudem, dass alle eingereichten Fassungen übereinstimmen.

Esslingen, 22.10.2025

---

Tim Jauch

---

Annabel Heberle

## Inhaltsverzeichnis

<b>1 Einleitung</b>	<b>1</b>
1.1 Vorstellung des Projekts	1
1.2 Zielsetzung und Relevanz	1
1.3 Aufbau der Arbeit	1
<b>2 Projektbeschreibung</b>	<b>2</b>
2.1 Problemstellung und Anforderungen	2
2.2 Zielgruppe und Personas	2
<b>3 Softwarearchitektur und Technologie</b>	<b>3</b>
3.1 Gesamtübersicht der Architektur	3
3.2 Frontend/Benutzeroberfläche	3
3.3 Backend, Datenbank und Schnittstellen	3
3.4 LLM-Integration (MCP, Wrapper, API)	3
3.5 Sicherheits- und Authentifizierungsmechanismen	3
<b>4 Implementierung</b>	<b>4</b>
4.1 Vorgehensmodell und Projektmanagement	4
4.2 Code-Generierung und Entwicklung	4
4.3 Datenmodelle und Schnittstellenimplementierung	4
4.4 Versionskontrolle, Build- und CI-Prozesse	4
<b>5 Einsatz der LLMs und Prompt Engineering</b>	<b>5</b>
5.1 Auswahl der LLMs und Dienste	5
5.2 Aufbau und Dokumentation der Prompts	5
5.3 Generierte Assets (Code, Bilder, Audio, Video, Text)	5
5.4 Fehleranalyse und Optimierung	5
5.5 Qualitätssicherung bei der LLM-Nutzung	5
<b>6 Erfahrung, Herausforderungen und Reflexion</b>	<b>6</b>
6.1 Positive Erfahrungen und Erfolgsgeschichten	6
6.2 Schwierigkeiten und Problemstellung	6
6.3 Grenzen und Risiken von LLMs	6
6.4 Lessons Learned und Empfehlungen	6
6.5 Zukunft der LLM-gestützten Entwicklung	6
<b>7 Zusammenfassung und Fazit</b>	<b>7</b>
7.1 Zusammenfassung der Arbeit	7
7.2 Persönliche Reflexion und Fazit	7

---

<b>Literatur .....</b>	<b>a</b>
------------------------	----------

## Abkürzungsverzeichnis

<b>API</b>	Application Programming Interface
<b>HTTP</b>	Hypertext Transfer Protocol
<b>REST</b>	Representational State Transfer

## Glossar

<b>Exploit</b>	An exploit is a method or piece of code that takes advantage of vulnerabilities in software, applications, networks, operating systems, or hardware, typically for malicious purposes.
<b>Patch</b>	A patch is data that is intended to be used to modify an existing software resource such as a program or a file, often to fix bugs and security vulnerabilities.
<b>Vulnerability</b>	A Vulnerability is a flaw in a computer system that weakens the overall security of the system.

# **1 Einleitung**

Stack Attack: All or nothing!

## **1.1 Vorstellung des Projekts**

Stack Attack ist ein Softwareprojekt, welches mit Hilfe von LLMs entwickelt und dokumentiert wird. Das Ziel des Projekts ist es, eine Online Version des beliebten Quartett Kartenspiels zu erstellen. Hierbei gilt es den Aufwand bei der Erstellung der Software mithilfe von LLMs zu minimieren und gleichzeitig eine qualitativ hochwertige Software zu entwickeln.

## **1.2 Zielsetzung und Relevanz**

## **1.3 Aufbau der Arbeit**

## **2 Projektbeschreibung**

### **2.1 Problemstellung und Anforderungen**

### **2.2 Zielgruppe und Personas**



## **3 Softwarearchitektur und Technologie**

### **3.1 Gesamtübersicht der Architektur**

### **3.2 Frontend/Benutzeroberfläche**

### **3.3 Backend, Datenbank und Schnittstellen**

### **3.4 LLM-Integration (MCP, Wrapper, API)**

### **3.5 Sicherheits- und Authentifizierungsmechanismen**

## **4 Implementierung**

### **4.1 Vorgehensmodell und Projektmanagement**

### **4.2 Code-Generierung und Entwicklung**

### **4.3 Datenmodelle und Schnittstellenimplementierung**

### **4.4 Versionskontrolle, Build- und CI-Prozesse**

## **5 Einsatz der LLMs und Prompt Engineering**

### **5.1 Auswahl der LLMs und Dienste**

### **5.2 Aufbau und Dokumentation der Prompts**

### **5.3 Generierte Assets (Code, Bilder, Audio, Video, Text)**

### **5.4 Fehleranalyse und Optimierung**

### **5.5 Qualitätssicherung bei der LLM-Nutzung**

## **6 Erfahrung, Herausforderungen und Reflexion**

### **6.1 Positive Erfahrungen und Erfolgsgeschichten**

### **6.2 Schwierigkeiten und Problemstellung**

### **6.3 Grenzen und Risiken von LLMs**

### **6.4 Lessons Learned und Empfehlungen**

### **6.5 Zukunft der LLM-gestützten Entwicklung**

## **7 Zusammenfassung und Fazit**

### **7.1 Zusammenfassung der Arbeit**

### **7.2 Persönliche Reflexion und Fazit**

## Literatur