

Gliederung

1. Einleitung

1.1 Projektziel und -aufbau

2. Positiven und Negativen Erfahrungen

2.1 Positive Erfahrungen

2.2 Negativen Erfahrungen

3. Sicherheit und Vertrauen in KI-generierte Anwendungen(Meine Bedenken und Perspektiven)

4. Fazit und Ausblick

Erfahrungsbericht: GitHub Copilot in der App-Entwicklung

1. Einleitung

In der heutigen digitalen Welt ist die Fähigkeit, Software schnell und effizient zu entwickeln, von entscheidender Bedeutung. Die Ansprüche an Softwareentwickler steigen, da Unternehmen und Benutzer innovative und benutzerfreundliche Anwendungen erwarten. In diesem Kontext haben sich KI-gestützte Tools wie GitHub Copilot als nützliche Helfer herausgestellt. Dieses Tool verspricht, den Entwicklungsprozess zu verbessern und die Qualität des Codes zu verbessern. In meinem Bericht teile ich meine Erfahrungen bei der Entwicklung einer App zur Buchverwaltung, die Kauf-, Verkaufs- und Ausleih Funktionen bietet. Ich werde die Stärken und Schwächen von GitHub Copilot beleuchten und meine Gedanken zur Sicherheit von KI-generierten Anwendungen äußern.

1.1 Projektziel und -aufbau

Mein Ziel war es, eine App zu entwickeln, mit der Nutzer Bücher kaufen und ausleihen können. Ich wollte eine Plattform schaffen, bei der die verschiedenen Nutzergruppen klar getrennt sind und die leicht zu benutzen ist. Dafür war eine gute Struktur der App nötig, die drei Hauptgruppen berücksichtigt. Die erste Gruppe sind die Administratoren. Sie sollten wichtige Verwaltungsfunktionen haben, zum Beispiel die Möglichkeit, Benutzer zu verwalten, Buchbestände zu aktualisieren und Berichte über Verkäufe und das Nutzerverhalten zu erstellen. Diese Funktionen sind wichtig, damit die App gut funktioniert. Die zweite Gruppe sind die Mitarbeiter. Sie sollten Bestände einsehen, Transaktionen bearbeiten und Kundenanfragen verwalten können. Für sie war es wichtig, dass alles gut läuft und die Kunden zufrieden sind. Die dritte Gruppe sind die Kunden. Für sie wollte ich eine einfache Möglichkeit schaffen, nach Büchern zu suchen, sie zu kaufen oder auszuleihen. Damit alle diese Anforderungen erfüllt werden, musste die App eine zuverlässige Verbindung zur Datenbank haben. Sie sollte alle Informationen über Nutzer, Bücher und Transaktionen gut verwalten, damit alles reibungslos läuft und die Nutzer zufrieden sind.

2. Positiven und Negativen Erfahrungen

In diesem Abschnitt möchte ich meine persönlichen Erfahrungen mit GitHub Copilot während der Entwicklung meiner App zur Buchverwaltung teilen. Dabei werde ich sowohl die positiven als auch die negativen Aspekte des Tools beleuchten. GitHub Copilot hat mir einerseits viele Erleichterungen und nützliche Hilfestellungen geboten, andererseits gab es auch Herausforderungen, die mich zum Nachdenken über die Grenzen der KI in der Softwareentwicklung angeregt haben. Im Folgenden

werde ich sowohl die Stärken als auch die Schwächen dieses Tools anhand meiner eigenen Erfahrungen darstellen.

2.1 Positive Erfahrungen

Zu Beginn meiner Arbeit mit GitHub Copilot war ich richtig begeistert von den Möglichkeiten, die das Tool bietet. Die Vorstellung, dass eine KI Code generieren kann und mir beim Programmieren direkt Vorschläge macht, fand ich total spannend. Also begann ich damit, die Grundfunktionen meiner App zu entwickeln und einfach mal zu testen, was Copilot vorschlägt. Es war beeindruckend, wie oft der vorgeschlagene Code tatsächlich genau das war, was ich gerade brauchte.

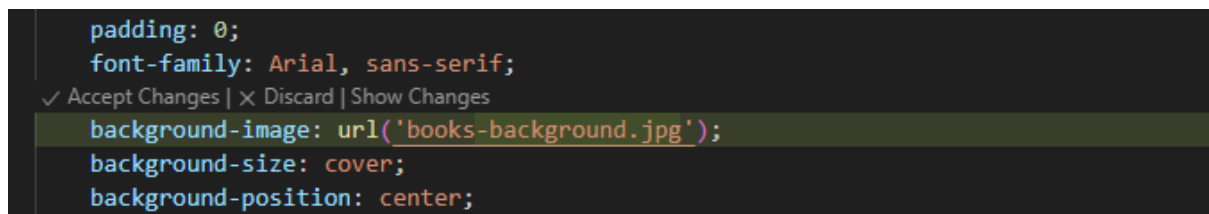


Bild 1: Änderung anzeigen

Im Bild 1 finde ich es jedoch gut, dass Copilot bei Änderungen am Code anzeigt, was geändert wurde, und mir die Möglichkeit gibt, diese Änderungen zu akzeptieren oder abzulehnen.

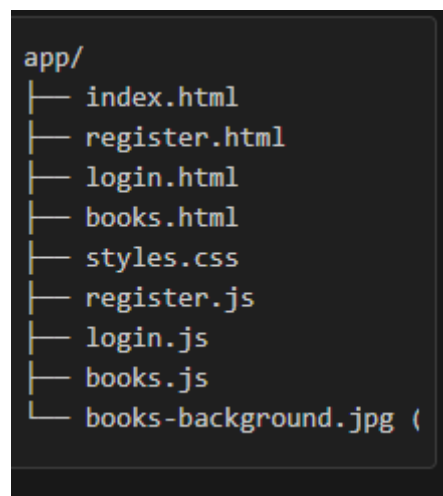


Bild 2: Projektstruktur

Im Bild 2 finde ich es sehr hilfreich, dass Copilot mir zeigt, welche Änderungen vorgenommen wurden. So kann ich sicherstellen, dass meine Projektstruktur korrekt aufgebaut ist und das Projekt reibungslos läuft.

sicherstellen, dass Node.js und npm korrekt installiert sind und dass wir die richtigen Schritte befolgen, um das Projekt zum Laufen zu bringen.

Schritt 1: Node.js und npm installieren

1. Node.js-Installer herunterladen:

- Gehen Sie zur [Node.js-Download-Seite](#).
- Laden Sie den Windows-Installer für die LTS-Version (Long Term Support) herunter.

2. Installer ausführen:

- Führen Sie den heruntergeladenen Installer aus.
- Folgen Sie den Installationsanweisungen. Stellen Sie sicher, dass das Kontrollkästchen "Add to PATH" (Zum PATH hinzufügen) während des Installationsprozesses aktiviert ist.

Schritt 2: Überprüfen Sie die Installation

1. Öffnen Sie ein neues

Eingabeaufforderungs- oder PowerShell-Fenster.

2. Führen Sie die folgenden Befehle aus, um zu überprüfen, ob Node.js und npm korrekt installiert sind:

```
node -v  
npm -v
```

- Sie sollten die Versionsnummern von Node.js und npm sehen.

[How can I add an image to the homepage of my website?](#)

Bild 3: Projektstruktur

Im Bild 3 gefällt mir besonders, dass Copilot nicht nur beim Codieren hilft, sondern auch bei technischen Problemen unterstützen kann. Ehrlich gesagt, war das oft sehr hilfreich – zum Beispiel hat er mir gezeigt, wie ich npm auf meinem Rechner installieren muss und wo ich es einfügen soll.

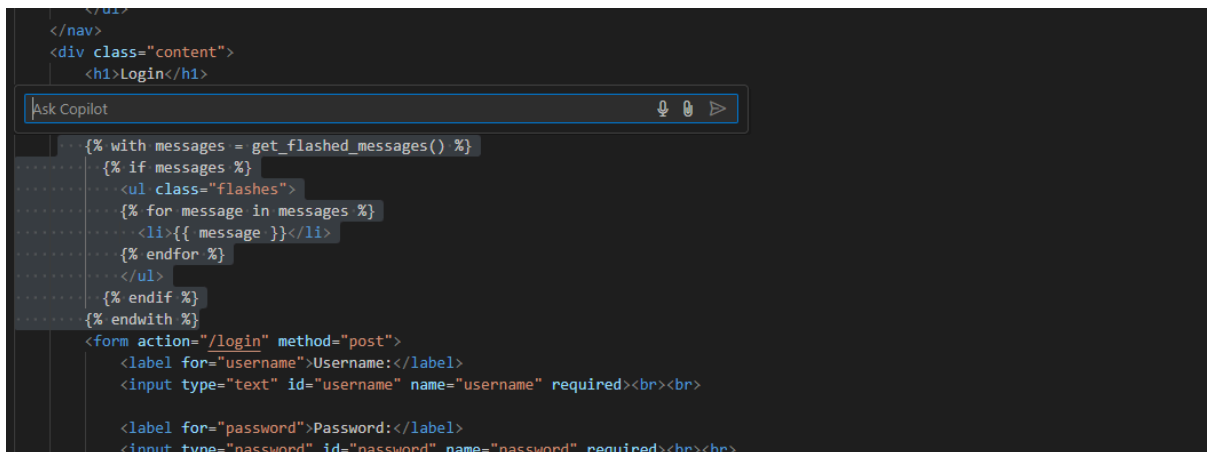


Bild 4: bestimmten Codeabschnitt auswählen

Bei der vierten Abbildung kann man einen bestimmten Codeabschnitt auswählen und Copilot bitten, ihn zu reparieren oder Änderungen vorzunehmen. Das fand ich wirklich hilfreich.

2.2 Negativen Erfahrungen

Trotz dieser positiven Erfahrungen merkte ich jedoch bald, dass GitHub Copilot auch seine Herausforderungen hat. Die größte Hürde war die Integration der Datenbank. Copilot schlug mir zwar Code für die Datenbankabfragen vor, aber ich stieß immer wieder auf Verbindungsprobleme, die sich nicht einfach lösen ließen. Ich musste feststellen, dass der von Copilot generierte Code zwar oft syntaktisch richtig war, aber in der Praxis nicht immer funktionierte. Nach paar Stunden des Ausprobierens und Anpassens wurde mir klar, dass ich die Datenbankverbindung einfach nicht hinbekomme. Trotz zahlreicher Anpassungen blieben die Verbindungsprobleme bestehen, was mich schließlich dazu zwang, mein Projektziel zu ändern. Copilot schlug sogar irgendwann vor, dass ich ein Kommando in der Eingabeaufforderung (CMD) nutzen könnte, aber da dies mein erstes Projekt war und ich den Verlauf des Chats nicht mehr hatte, konnte ich hier leider keine weiteren Details oder Bilder zu dem Problem anbieten. Als ich dann das Projektziel änderte, merkte ich, dass Copilot sich weiterhin an den alten Anforderungen orientierte und dazu Vorschläge machte. Diese Mischung aus alten und neuen Ideen sorgte für zusätzliche Verwirrung und machte die Entwicklung langsamer, weil ich oft nicht mehr sicher war, welche Vorschläge noch sinnvoll waren. Ein Nachteil ist, dass, wenn ich zum Beispiel eine Aufgabe formuliere und diese im Copilot-Chat schreibe, dann den Ordner ändere und wieder öffne, alle geschriebenen Sätze im Chat gelöscht werden. Wenn ich dann den Satz erneut eingebe, wird ein komplett neuer Code generiert, und ich muss alles von vorne schreiben.

```

# app/__init__.py
from flask import Flask
from flask_sqlalchemy import SQLAlchemy
from flask_login import LoginManager

db = SQLAlchemy()
login_manager = LoginManager()

def create_app():
    app = Flask(__name__)
    app.config.from_object('config.Config')

    db.init_app(app)
    login_manager.init_app(app)

    from .auth import auth as auth_blueprint
    app.register_blueprint(auth_blueprint)

    from .books import books as books_blueprint
    app.register_blueprint(books_blueprint)

    from .dashboard import dashboard as dashboard_blueprint
    app.register_blueprint(dashboard_blueprint)

    return app

# config.py
class Config:
    SECRET_KEY = 'your_secret_key'
    SQLALCHEMY_DATABASE_URI = 'sqlite:///site.db'
    SQLALCHEMY_TRACK_MODIFICATIONS = False

# app/models.py
from . import db
from flask_login import UserMixin

class User(UserMixin, db.Model):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(150), unique=True, nullable=False)
    email = db.Column(db.String(150), unique=True, nullable=False)
    password = db.Column(db.String(150), nullable=False)

class Book(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    title = db.Column(db.String(150), nullable=False)
    author = db.Column(db.String(150), nullable=False)
    price = db.Column(db.Float, nullable=False)

```

Bild 5: Ohne Kommentare

Das Bild zeigt einen weiteren Nachteil, den ich entdeckt habe: Wenn Copilot Code generiert, tut er dies ohne Kommentare, es sei denn, ich fordere ihn ausdrücklich dazu auf. Die Kommentare sind jedoch wichtig, da sie es erleichtern, den Code nachzuvollziehen und bestimmte Teile schnell zu finden, wenn man nach bestimmten Funktionen sucht.

```
<li><a href="#services">Services</a></li>
<li><a href="#contact">Contact</a></li>
</ul>
</nav>
<div class="content">
  <h1>Welcome to the Book Page</h1>
  <p>This is a sample page with a books backgr
</div>
</body>
</html>

styles.css

body {
  margin: 0;
  padding: 0;
  font-family: Arial, sans-serif;
  background-image: url("books-background.jpg");
  background-size: cover;
  background-position: center;

3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Books Background Page</title>
7   <link rel="stylesheet" href="styles.css">
8 </head>
9 <body>
10   <div class="content">
11     <h1>Welcome to the Book Page</h1>
12     <p>This is a sample page with a books background.</p>
13   </div>
14 </body>
15 </html>

index.html
9 <body>
10+   <nav>
11+     <ul>
12+       <li><a href="#home">Home</a></li>
13+       <li><a href="#about">About</a></li>
14+       <li><a href="#services">Services</a></li>
15+       <li><a href="#contact">Contact</a></li>
16+     </ul>
17+   </nav>
18   <div class="content">

background-repeat: no-repeat;
height: 100vh;
overflow: hidden; /* Prevents scrolling */

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
PS C:\Users\Teilnehmer\Desktop\app>
```

Bild 6: Änderungen akzeptieren und ablehnen

Ich hatte das Problem, dass ich ein Bild als Hintergrund einfügen wollte. Copilot hat den Code geschrieben, aber es ist nichts passiert – das Bild war nicht sichtbar. Ich habe das im Chat erklärt, und er hat den Code erneut geschrieben, sodass das Bild sichtbar wurde, aber es war dupliziert. Nachdem ich es erneut erklärt habe, hat Copilot den Code diesmal korrekt geschrieben. Allerdings wurde die Funktion, die Änderungen am Code anzeigt und es mir ermöglicht, diese zu akzeptieren oder abzulehnen, nicht mehr angezeigt. Ich habe versucht, diese Funktion wieder zu aktivieren, aber das hat nicht funktioniert, bis ich den Chat beendet und neu gestartet habe.

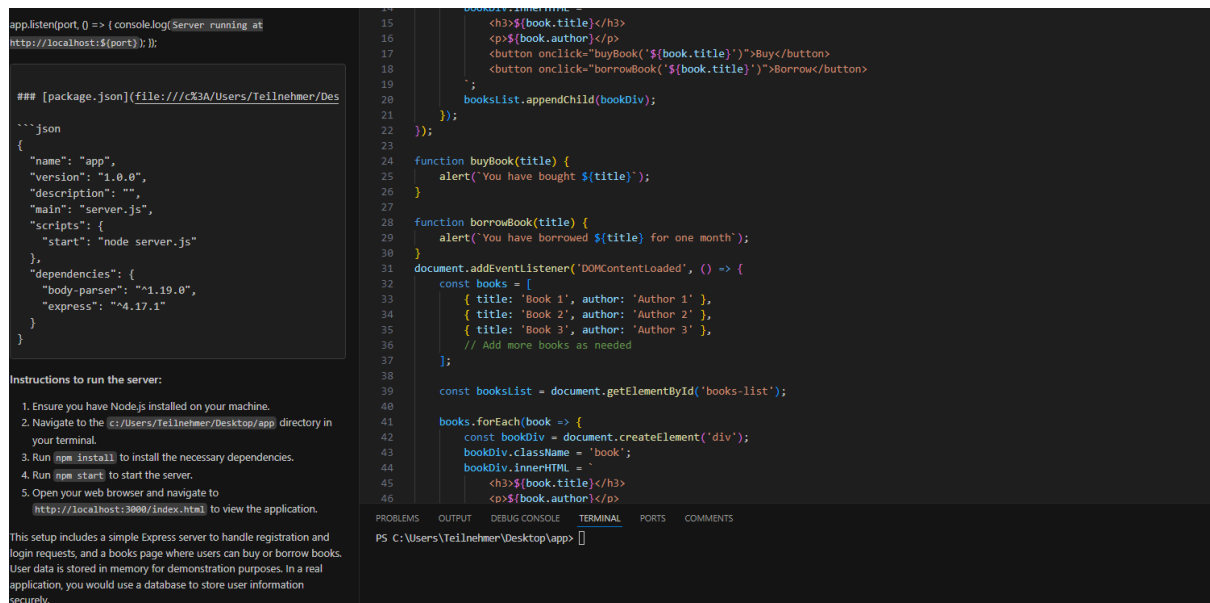


Bild 7: keine Möglichkeit hinzufügen

Im Bild 7 hatte ich das Problem, dass ich keinen Code mehr hinzufügen konnte und keine Möglichkeit hatte, neuen Code einzufügen, außer ihn zu kopieren und manuell einzufügen. Abgesehen davon funktionierte das Hinzufügen von Code nicht.

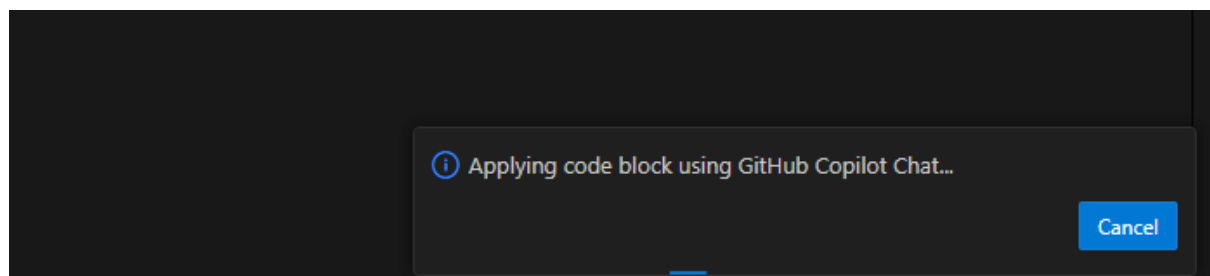


Bild 8: Rückmeldung zu bekommen

Im Bild 8 gibt es kein konkretes Problem, aber ich würde mir wünschen, dass nach dem Anwenden des Codes eine Nachricht erscheint, die bestätigt, dass der Code erfolgreich angewendet wurde. Es ist frustrierend, warten zu müssen, ohne eine Rückmeldung zu bekommen. Wenn sich nichts ändert und keine Nachricht angezeigt wird, könnte man leicht denken, dass der Prozess nicht erfolgreich war.



Bild 9: der gesamte Code aktualisiert

Im Bild 9 wird bei jeder Frage an Copilot der gesamte Code aktualisiert, und ich muss alles erneut anwenden, obwohl kein neuer Code hinzugekommen ist. Das finde ich ziemlich unnötig und störend.

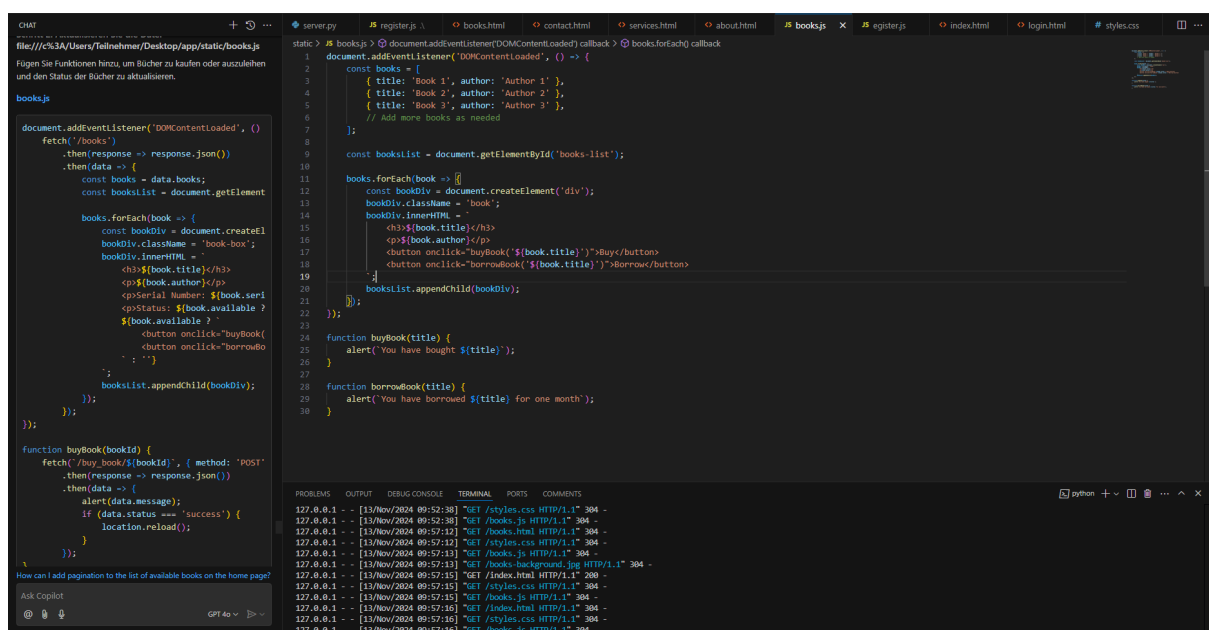


Bild 10: Prozess nicht ausgeführt

Im Bild 10, wenn ich auf den „Apply“-Button drücke, erhalte ich die Nachricht, dass der Prozess ausgeführt wird, aber der neue Code wird nicht angewendet! Ich musste alles löschen und den Code manuell per Copy-Paste einfügen.

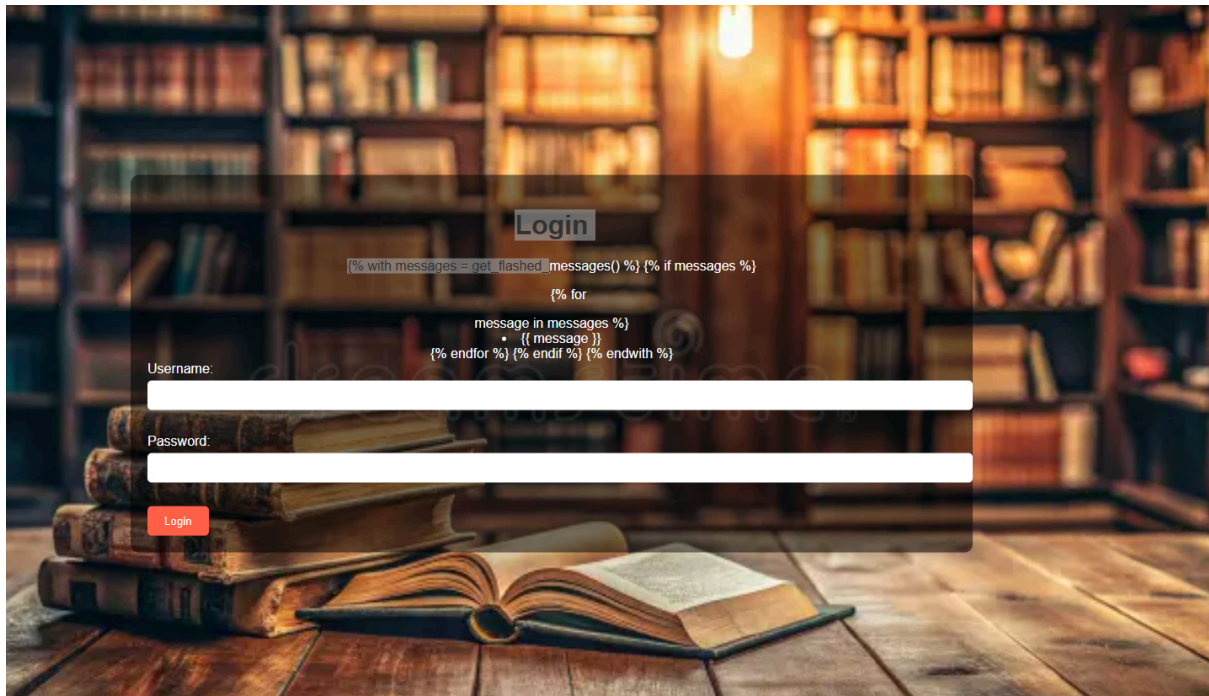


Bild 11: code problem

Im Bild 11 hatte ich das Problem, dass ich Copilot gebeten habe, dafür zu sorgen, dass der Benutzer eine Nachricht erhält und zur Anmeldeseite weitergeleitet wird, wenn er einen falschen Benutzernamen oder ein falsches Passwort eingibt, anstatt die Nachricht auf einer leeren Seite anzuzeigen. Copilot hat das zwar umgesetzt, aber dabei ein anderes Problem verursacht. Selbst nach mehreren Versuchen, ihn zur Korrektur aufzufordern, hat es leider nicht geklappt.

3. Sicherheit und Vertrauen in KI-generierte Anwendungen(Meine Bedenken und Perspektiven)

Während meines Projekts habe ich auch über die Sicherheit von KI-generierten Anwendungen nachgedacht. Ich bin der Meinung, dass es mit KI-Tools wie GitHub Copilot möglich ist, auch ohne tiefgehende Programmierkenntnisse einfache Apps oder Webseiten zu erstellen. Allerdings habe ich ernsthafte Bedenken, wenn es um die Sicherheit solcher Anwendungen geht. Ein großes Risiko sehe ich, wenn Nutzer zum Beispiel ihre Kreditkarteninformationen in eine App eingeben. In diesem Fall muss die Anwendung besonders sicher sein. Ein unsicherer Code kann dazu führen, dass Daten missbraucht werden oder es zu Betrug kommt. Deshalb halte ich es für unerlässlich, dass solche Anwendungen von Fachleuten geprüft werden, bevor sie live gehen. Auch der Datenschutz ist ein wichtiger Punkt. Bei Apps, die persönliche

Informationen sammeln, müssen die Datenschutzgesetze beachtet werden. Entwickler sollten sicherstellen, dass sie diese gesetzlichen Vorgaben einhalten und Maßnahmen zum Schutz der Benutzerdaten treffen.

4. Fazit und Ausblick

Abschließend lässt sich sagen, dass GitHub Copilot für mich eine wertvolle Hilfe bei der Entwicklung meiner App war, insbesondere bei der Codegenerierung und der Unterstützung bei technischen Problemen. Die Möglichkeit, schnelle Vorschläge zu erhalten, hat den Entwicklungsprozess beschleunigt und mir geholfen, Fehler zu vermeiden. Doch trotz dieser positiven Erfahrungen gab es auch Herausforderungen. Besonders die Integration der Datenbank und einige technische Probleme haben mich Zeit und Geduld gekostet. Auch die Tatsache, dass Copilot bei Änderungen am Code keine Kommentare hinzufügt, war für mich ein Nachteil, da diese für das Verständnis und die Wartung des Codes sehr hilfreich sind. Ein weiteres wichtiges Thema, das ich während meines Projekts reflektiert habe, ist die Sicherheit von KI-generierten Anwendungen. Während Copilot es mir ermöglicht hat, eine funktionierende App zu erstellen, muss ich betonen, dass besonders bei sicherheitskritischen Anwendungen die Qualität und Sicherheit des generierten Codes unbedingt von Fachleuten überprüft werden sollten. Ohne diese Überprüfung können Sicherheitslücken entstehen, die die Nutzer gefährden könnten. Auch der Datenschutz ist ein zentraler Aspekt, der nicht vernachlässigt werden darf. Für die Zukunft sehe ich großes Potenzial in der Weiterentwicklung von KI-Tools wie GitHub Copilot. Wenn diese Tools weiter optimiert werden und vor allem bei der Codequalität und Sicherheit verlässlicher werden, könnten sie in der Softwareentwicklung eine noch größere Rolle spielen. Ich hoffe, dass zukünftige Updates auch die von mir angesprochenen Probleme, wie etwa das Fehlen von Kommentaren im Code und die zuverlässige Funktionsweise, behoben werden. Insgesamt bleibt KI ein mächtiges Werkzeug, das die Entwicklung von Software vereinfacht, aber nicht die Verantwortung der Entwickler ersetzt, besonders in Bereichen wie Sicherheit und Datenschutz.