

# **Benutzerschnittstellen entwerfen und implementieren**

## Fallstudie und die Technische Dokumentation

# Inhalt

Organisationsteil.....	3
Projektorganisation .....	3
Zielgruppe.....	3
Dateiablage.....	3
Datensicherung.....	3
Arbeitsweise/Arbeitsbestätigung.....	3
Arbeitsjournal.....	4
Reflektion.....	7
Fazit zum CsBe-Noten Projekt .....	7
Persönliches Fazit .....	7
Tabellenverzeichnis.....	7
Abbildungsverzeichnis .....	7
Projektteil .....	8
Analyse .....	8
Ausgangslage (ist-Situation) .....	8
Projektziel (soll-Situation) .....	8
1: Authentifizierung.....	8
2: Stammdaten-Erfassung .....	8
3: Fach- und Modulauswahl .....	9
4: Daten zu Noten und Absenzen .....	9
5: Übersicht und Analyse.....	9
6: Benutzerfreundlichkeit und Barrierefreiheit .....	9
7: Mehrsprachigkeit.....	9
8: Sicherer Logout.....	9
Anforderungen .....	10
User Stories .....	11
Storyboard .....	12
Umsetzung.....	14
Struktur.....	14
Testing.....	14
Testkonzept.....	14
Testfälle.....	14
Referenzen.....	15

# Organisationsteil

## Projektorganisation

### Zielgruppe

Die Schule CsBe möchte den Studierenden ein Tool in die Hand geben, mit dem sie die eigenen Noten und Absenzen verwalten können. Ziel soll dabei sein, dass die Lernenden ihren aktuellen Leistungsstand selbst beurteilen können.

### Dateiablage

Die Projektbezogenen Dateien werden in zwei Gruppen aufgeteilt.

- Planungs- und Dokumentationsrelevante Dateien
- Projektdateien

### Datensicherung

Alle projektbezogene Dateien werden auf dem Rechner gespeichert weil die VM zu oft abstürzt und man es auf dem nicht virtuellen Speichermedium hat.

Manuelles Backup auf einer externen Festplatte. Am Ende jeden Arbeitstages wird der gesamte Inhalt des Projektordners (Dokumentation und Workspace) auf eine externen Festplatte kopiert. Dabei wird das aktuelle Datum in den Ordernamen geschrieben, damit eine übersichtliche Versionierung entstanden.

## Arbeitsweise/Arbeitsbestätigung

Ich Stefan Mitric habe Herr Giulio Iannattone gefragt ob ich mit HTML, CSS und JavaScript arbeiten können. Er gab mir Bescheid das ich das benutzen darf um meinen Projekt zu machen

# Arbeitsjournal

## Arbeitsjournal: Tag 1

Tabelle 1: Arbeitsjournal Tag 1

Tätigkeiten	Aufwand geplant (Std)	Aufwand effektiv (Std)
Konzept erstellt. In draw.io erstellt	1h	1.5h
Alle Fenster erstellt die ich brauche.	3h	3h
Anfangen mit Login Seite.	1h	2h
<b>Total:</b>	<b>5h</b>	<b>6.5h</b>
<b>Abweichung SOLL / IST:</b>	<b>1.5h</b>	
<b>Probleme</b>		
Platzierung der Text Felder hatte paar Probleme.		
<b>Hilfestellungen</b>		
Keine		
<b>Reflexion</b>		
Hat alles funktioniert wie ich geplant habe.		
<b>Nächste Schritte</b>		
Buttons einfügen um auf die anderen Fenstern zu gelangen.		

## Konzept für die Website

The wireframe illustrates a web application for managing grades (Notenverwaltung). It consists of three main interface components:

- Login/Registration Screen:** Titled "CSBE NOTENVERWALTUNG", it features input fields for "Username" and "Passwort", a "LOGIN" button, and links for "Registrieren" and "Passwort vergessen".
- User Profile Screen:** Displays the user's name "Stefan Mitric CSBE BIS22" and includes two "Noten Hinzufügen" buttons.
- Data Entry Screen:** Contains dropdown menus for "Semester" and "Module", a table with columns for "Note" and "Prozent", and a "Hinzufügen Buttons" column with plus icons for adding new entries. A "BACK" button is located at the bottom.

## Arbeitsjournal: Tag 2

Tabelle 2:Arbeitsjournal Tag 2

Tätigkeiten	Aufwand geplant (Std)	Aufwand effektiv (Std)
Buttons erstellt.	1h	1h
JS erstellt damit man mit dem Augen Icon Button das Passwort anschauen kann und wieder verdecken kann.	3h	3h
Tabelle erstellt auf der Notenübersicht Fenster.	1h	2h
UI angepasst	2h	2h
<b>Total:</b>	<b>7h</b>	<b>8h</b>
<b>Abweichung SOLL / IST:</b>	<b>1h</b>	
<b>Probleme</b>		
Keine		
<b>Hilfestellungen</b>		
Hilfestellung war GPT-4 für den JS code		
<b>Reflexion</b>		
Endlich habe ich alle Fenster auch verknüpfen können.		
<b>Nächste Schritte</b>		
Login Screen fertig gestalten.		

## Arbeitsjournal: Tag 3

Tabelle 3:Arbeitsjournal Tag 3

Tätigkeiten	Aufwand geplant (Std)	Aufwand effektiv (Std)
Login Screen komplett fertig gemacht mit allen Buttons und Sprachen	2h	2h
Habe die Notenübersicht auch fertig gemacht mit allen Buttons und Sprachen.	2h	1h
Habe die Noteneintrag Fenster fertig gemacht und alle Buttons so- wie alle sprachen eingefügt.	2h	1h
Total:	6h	4h
Abweichung SOLL / IST:	2h	
Probleme		
Hatte keine Probleme		
Hilfestellungen		
Hilfestellung war GPT-4 zum Kommentieren		
Reflexion		
Ich habe alle Funktionen, die ich ursprünglich geplant hatte, erfolgreich implementieren können. Zudem ist die Benutzeroberfläche (UI) nun vollständig und einsatzbereit.		
Nächste Schritte		
Tests durchführen und Vielleicht UI noch ausbessern.		

## Arbeitsjournal: Tag 4

Tabelle 4:Arbeitsjournal Tag 4

Tätigkeiten	Aufwand geplant (Std)	Aufwand effektiv (Std)
Tests durchgeführt und das UI finalisiert.	1h	1h
<b>Total:</b>	<b>1h</b>	<b>1h</b>
<b>Abweichung SOLL / IST:</b>	<b>0h</b>	

## Reflektion

### Fazit zum CsBe-Noten Projekt

Das Projekt für die CsBe Schule stellt eine umfassende Herausforderung dar, sowohl in Bezug auf funktionale als auch nichtfunktionale Anforderungen. Es handelt sich um ein komplexes Vorhaben, das die Entwicklung einer Noten- und Absenzen Verwaltung für Studierende umfasst. Das Hauptziel besteht darin, ein Werkzeug zu schaffen, mit dem Studierende ihren akademischen Fortschritt selbst überwachen und verwalten können.

### Persönliches Fazit

Ich persönlich finde, dass ich für ein solches Projekt zu wenig Zeit erhalten habe. Etwas so Kurzfristiges zu erstellen, bei dem auch die Benutzerfreundlichkeit stimmen muss, fand ich recht knapp bemessen. Ansonsten hat es Spaß gemacht, wieder einmal mit HTML, CSS und JavaScript zu arbeiten. Für das nächste Mal würde ich mir einfach wünschen, mehr Zeit zu bekommen.

## Tabellenverzeichnis

Tabelle 1:Arbeitsjournal Tag 1.....	4
Tabelle 2:Arbeitsjournal Tag 2.....	5
Tabelle 3:Arbeitsjournal Tag 3.....	6
Tabelle 4:Arbeitsjournal Tag 4.....	7
Tabelle 6: Funktionale Anforderungen.....	10
Tabelle 7: Nichtfunktionale Anforderungen .....	11

## Abbildungsverzeichnis

Abbildung 1: Login Screen .....	12
Abbildung 2: Notenübersicht .....	13
Abbildung 3: Noteneintragen .....	13

# Projektteil

## Analyse

Sie erhalten den Auftrag, eine Fallstudie zu einer Problemstellung durchzuführen und die daraus resultierende Applikation objektbasiert umzusetzen. In dieser Fallstudie müssen Sie das gegebene Problem analysieren und eine geeignete Lösung entwerfen und modellieren. Es müssen Überlegungen zur Art der Umsetzung und zur Gestaltung der Benutzerschnittstelle angestellt werden.

### Ausgangslage (ist-Situation)

Die Schule CsBe möchte den Studierenden ein Tool in die Hand geben, mit dem sie die eigenen Noten und Absenzen verwalten können. Ziel soll dabei sein, dass die Lernenden ihren aktuellen Leistungsstand selbst beurteilen können.

Das Tool (CsBe-Noten) soll die Daten in einer DB ablegen. Diese Daten dürfen nur nach einer erfolgreichen Authentifizierung einsehbar sein. Die Nutzer:innen müssen zuerst ihre persönlichen Daten erfassen. Anschliessend können sie die belegten Fächer, Module und Semester auswählen oder neu erfassen. Die ÜK-Module sind speziell gekennzeichnet. Sobald ein Semester aktiviert wurde, kann die Benutzer:in einzelne Noten und Absenzen dem gewählten Semester und Fach eintragen. In einem Überblick kann für jedes Semester ein «Zeugnis» erzeugt werden. Dabei werden die Notendurchschnitte für die Fächer wie auch die Bereiche BKU und ÜK errechnet.

### Projektziel (soll-Situation)

Entwicklung einer sicheren, benutzerfreundlichen und barrierefreien Notenverwaltungssoftware für die Studierenden der Schule CsBe, um ihre akademischen Leistungen und Absenzen eigenständig zu verwalten und auszuwerten.

## Programm

### 1: Authentifizierung

1. Das Programm startet und prüft, ob bereits Nutzerdaten in der Datenbank vorhanden sind.
2. Wenn ja, muss der Benutzer sich authentifizieren, um Zugriff auf seine Daten zu erhalten.

### 2: Stammdaten-Erfassung

1. Ist die Datenbank leer, wird der Benutzer aufgefordert, seine Stammdaten einzugeben.
2. Hierzu gehören persönliche Informationen wie Name, Vorname, Geschlecht, Geburtsdatum, Ausbildungsgang, Benutzername, Passwort und Sprache.



### 3: Fach- und Modulauswahl

1. Nach der Erfassung der Stammdaten wählt der Benutzer die Fächer, Module und das aktuelle Semester aus.
2. Spezielle Module, z.B. ÜK-Module, werden entsprechend gekennzeichnet.

### 4: Daten zu Noten und Absenzen

1. Sobald ein Semester aktiviert ist, kann der Benutzer Noten und Absenzen für die jeweiligen Fächer und Module hinzufügen.
2. Noten werden als Dezimalzahlen zwischen 1,0 und 6,0 eingegeben.
3. Absenzen werden als ganze Zahlen zwischen 1 und 10 eingegeben.

### 5: Übersicht und Analyse

1. Auf der Übersichtsseite werden die Notendurchschnitte für die belegten Fächer und Bereiche angezeigt.
2. Die Absenzen werden sowohl für einzelne Fächer als auch insgesamt ausgewiesen.
3. Ein "Zeugnis" kann erstellt werden, das die Notendurchschnitte für das jeweilige Semester zusammenfasst.

### 6: Benutzerfreundlichkeit und Barrierefreiheit

1. Das Programm ist barrierefrei und für Screenreader optimiert.
2. Zusätzliche Features wie Soundeffekte und Piktogramme können für eine verbesserte Benutzererfahrung sorgen.

### 7: Mehrsprachigkeit

1. Die Software unterstützt mindestens drei Sprachen: Deutsch, Französisch und Englisch.

### 8: Sicherer Logout

1. Nach dem Logout sind keine persönlichen Daten mehr sichtbar und der Benutzer kehrt zur Authentifizierungsseite zurück.

### Dokumentation

1. Sicherheit und Datenschutz:
  - 1.1 Implementierung einer Authentifizierungsmechanismus, um sicherzustellen, dass nur berechtigte Benutzer auf ihre Daten zugreifen können.
2. Benutzerfreundlichkeit:
  - 2.1 Entwicklung einer intuitiven Benutzeroberfläche, die es den Benutzern ermöglicht, ihre persönlichen Daten, belegten Fächer, Module und Semester einfach zu erfassen und zu verwalten.
  - 2.2 Implementierung einer Funktion, die es ermöglicht, ein "Zeugnis" mit Notendurchschnitten für jedes Semester zu erzeugen.

3. Barrierefreiheit:
  - 3.1 Die Software muss barrierefrei sein und für die Bedienung mithilfe eines Screenreaders optimiert werden.
4. Multilingualität:
  - 4.1 Die Software muss in mindestens drei Sprachen (Deutsch, Französisch und Englisch) verfügbar sein.
5. Technische Anforderungen:
  - 5.1 Verwendung von Java Swing für die GUI und eines gängigen RDBMS für die Datenbank.
  - 5.2 Die Anwendung muss im 16:9 Seitenverhältnis (1920 x 1080) optimal dargestellt werden.
6. Dokumentation und Testing:
  - 6.1 Erstellung einer umfangreichen Dokumentation, die Use-Cases, Storyboards, Styleguides und Testprotokolle enthält.
  - 6.2 Durchführung von Usability-Tests und systematisches Abdecken aller Funktionen durch Testfälle.
7. Zusätzliche Benutzerfreundlichkeit:
  - 7.1 Implementierung von Soundeffekten, Piktogrammen und Sprachausgabe zur Verbesserung der Benutzererfahrung.
8. Termingerechte Abgabe:
  - 8.1 Die Software und alle begleitenden Dokumente müssen gemäß den Vorgaben der Schule fristgerecht eingereicht werden.

## Anforderungen

Dieser Abschnitt behandelt die konkreten Anforderungen an die zu erstellten Lösung.

*Tabelle 5: Funktionale Anforderungen*

	Anforderungen
1	<b>Authentifizierung:</b> Das Programm muss erkennen, ob bereits Nutzerdaten in der Datenbank gespeichert sind. Wenn ja, ist eine Authentifizierung erforderlich.
2	<b>Stammdatenerfassung:</b> Bei einer leeren Datenbank müssen zuerst die Stammdaten eingegeben werden. Diese enthalten persönliche Informationen und Fächer.
3	<b>Fach- und Modulauswahl:</b> Nutzer müssen ihre Fächer, Module und Semester auswählen oder neu erfassen können.
4	<b>Noten und Absenzen:</b> Die Nutzer sollen in der Lage sein, ihre Noten und Absenzen einzutragen.
5	<b>Übersichtsseite:</b> Eine Übersicht mit Notendurchschnitten und Absenzen muss verfügbar sein.
6	<b>Anpassbare Ansicht:</b> Die Möglichkeit, die Daten anderer Semester anzuzeigen, muss gegeben sein.
7	<b>Eingabeformat für Noten und Absenzen:</b> Noten als Dezimalzahlen von 1,0 bis 6,0 und Absenzen als ganze Zahlen von 1 bis 10.
8	<b>Aktualisierung der Übersichtsseite:</b> Nach einer neuen Eingabe muss die Übersicht aktualisiert werden.
9	<b>Sicherer Logout:</b> Nach dem Logout dürfen keine Nutzerdaten mehr sichtbar sein.

Tabelle 6: Nichtfunktionale Anforderungen

	Anforderungen
1	Technologie: <b>Umsetzung mit Java Swing.</b>
2	<b>Datenbank:</b> Verwendung eines gängigen RDBMS.
3	<b>Mehrsprachigkeit:</b> Mindestens Deutsch, Französisch und Englisch müssen unterstützt werden.
4	<b>Barrierefreiheit:</b> Das Programm muss für die Nutzung mit Screenreadern optimiert sein.
5	<b>Code-Qualität:</b> Der Code muss sauber strukturiert und gut kommentiert sein.

## User Stories

- **Authentifizierung**

1. Als Nutzer möchte ich mich authentifizieren können, um Zugriff auf meine persönlichen Daten zu erhalten.

- **Stammdatenerfassung**

1. Als neuer Nutzer möchte ich meine Stammdaten (Name, Vorname, Geburtsdatum, etc.) eingeben können, um ein neues Konto zu erstellen.

- **Fach- und Modulauswahl**

1. Als Nutzer möchte ich die Fächer, Module und Semester auswählen oder neu erfassen können, um meine Noten und Absenzen zuzuordnen.

- **Noten und Absenzen**

1. Als Nutzer möchte ich Noten zu den jeweiligen Fächern und Semestern hinzufügen können, um meinen Leistungsstand zu verfolgen.
2. Als Nutzer möchte ich Absenzen zu den jeweiligen Fächern und Semestern hinzufügen können, um meine Anwesenheit zu verfolgen.

- **Übersichtsseite**

1. Als Nutzer möchte ich eine Übersicht meiner Noten und Absenzen sehen können, um meinen aktuellen Leistungsstand schnell zu überblicken.

- **Anpassbare Ansicht**

1. Als Nutzer möchte ich die Daten verschiedener Semester sehen können, um meine Leistung im zeitlichen Verlauf zu betrachten.

- **Logout**

1. Als Nutzer möchte ich mich sicher abmelden können, um den Zugriff auf meine persönlichen Daten zu beschränken.

2.

- **Barrierefreiheit**

1. Als Nutzer mit eingeschränkter Sehfähigkeit möchte ich das Programm mit einem Screenreader nutzen können, um gleichberechtigten Zugang zu haben.

- **Mehrsprachigkeit**

1. Als nicht-deutschsprachiger Nutzer möchte ich das Programm in meiner Sprache (Französisch oder Englisch) verwenden können, um es besser zu verstehen.

## Storyboard

In diesem Abschnitt werden alle Fenster des Programms als Mockup dargestellt. Die effektive optische Gestaltung hängt vom auszuführenden System ab.

Alle Fenster des GUIs (mit Ausnahme der System-Dialoge) werden mit einer einheitlichen Grösse gestaltet. Der Benutzer kann die Fenstergrösse allerdings Belieben anpassen.

## Login Screen

Im Login-Bildschirm kann man verschiedene Sprachen auswählen, die dann im gesamten Programm beibehalten werden. Man hat die Möglichkeit, sich zu registrieren, allerdings ist diese Funktion noch nicht implementiert. Ebenso gibt es die Option, das Passwort zurückzusetzen, die ebenfalls noch nicht funktioniert. Die Anmeldefunktion ist jedoch implementiert. Hier kann man beliebige Daten eingeben, um Zugang zu erhalten. Dies liegt daran, dass die Registrierungsfunktion noch nicht umgesetzt ist. Beide Felder (Benutzername und Passwort) müssen jedoch ausgefüllt sein, um Zugang zu bekommen.

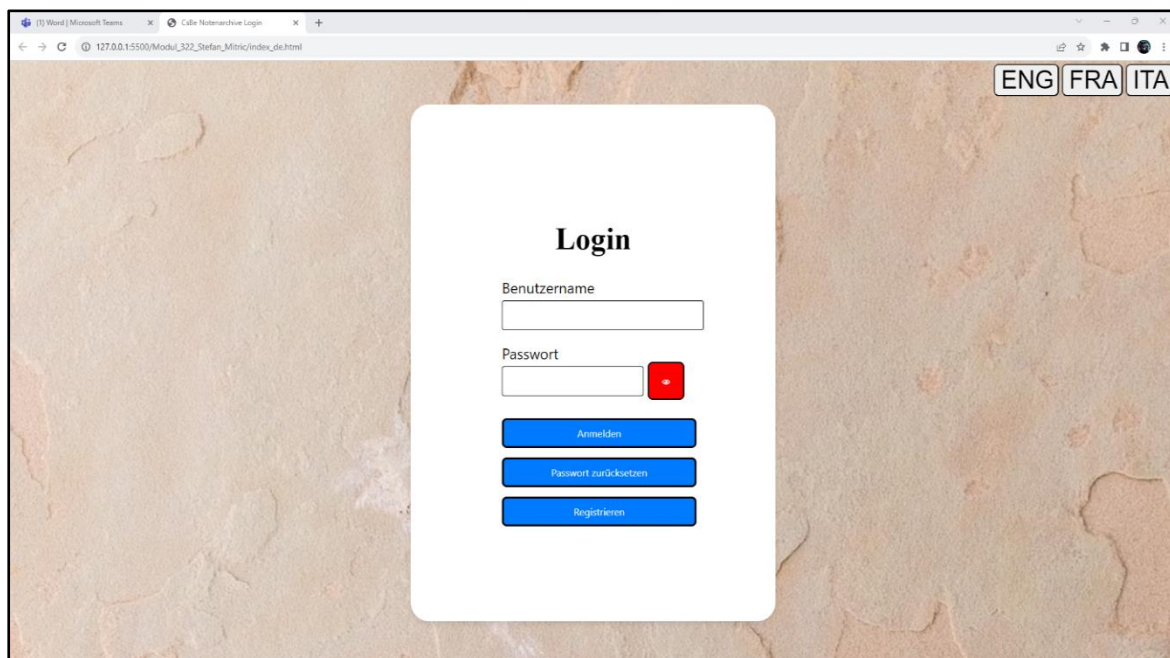


Abbildung 1: Login Screen

## Notenübersicht

In der Notenübersicht können die eingegebenen Noten betrachtet werden. Aufgrund von Zeitmangel konnte die Funktion jedoch nicht vollständig implementiert werden. Daher sind bereits vorgefertigte Ergebnisse vorhanden, die man direkt im JavaScript-Code ändern muss. Mit dem "Noteneintrag"-Button kann man zum nächsten Fenster wechseln, um dort die Noten einzugeben. Das Informations-Icon ("i") ist leider nicht funktional, da mir keine passenden Informationen eingefallen sind, die hier sinnvoll wären.

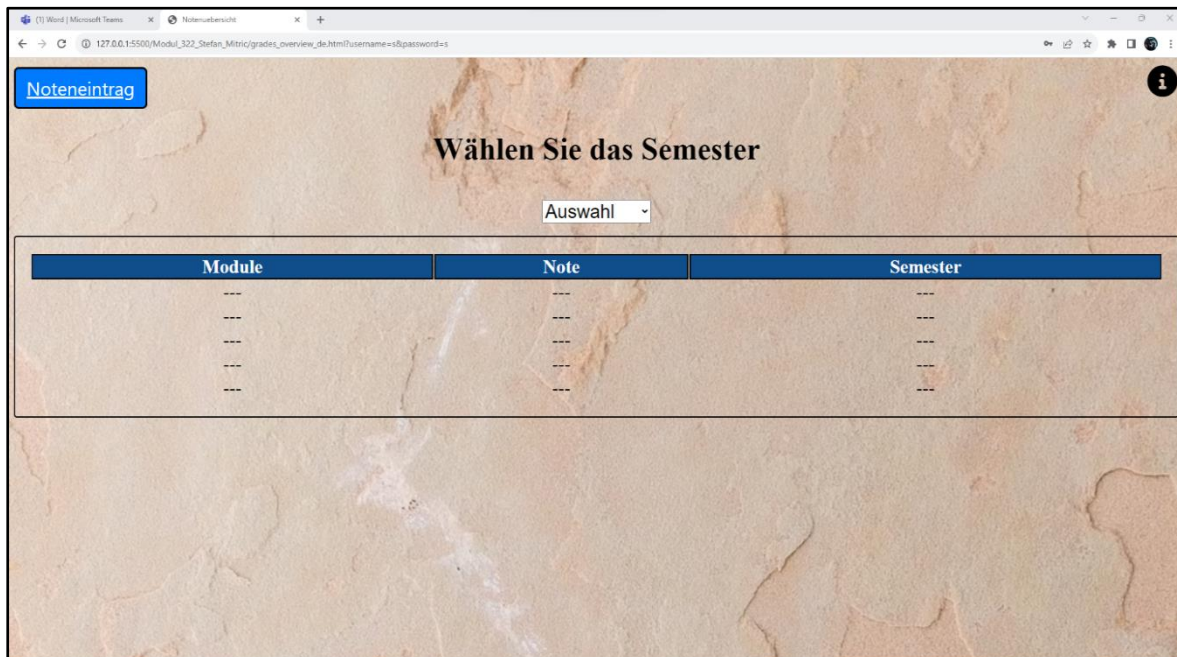


Abbildung 2: Notenübersicht

## Noteneinfügen

In diesem Fenster können die Module, Noten und das Semester eingegeben werden. Leider ist diese Funktion nicht aktiv. Mit dem "Zurück"-Button gelangt man wieder zur Notenübersicht.

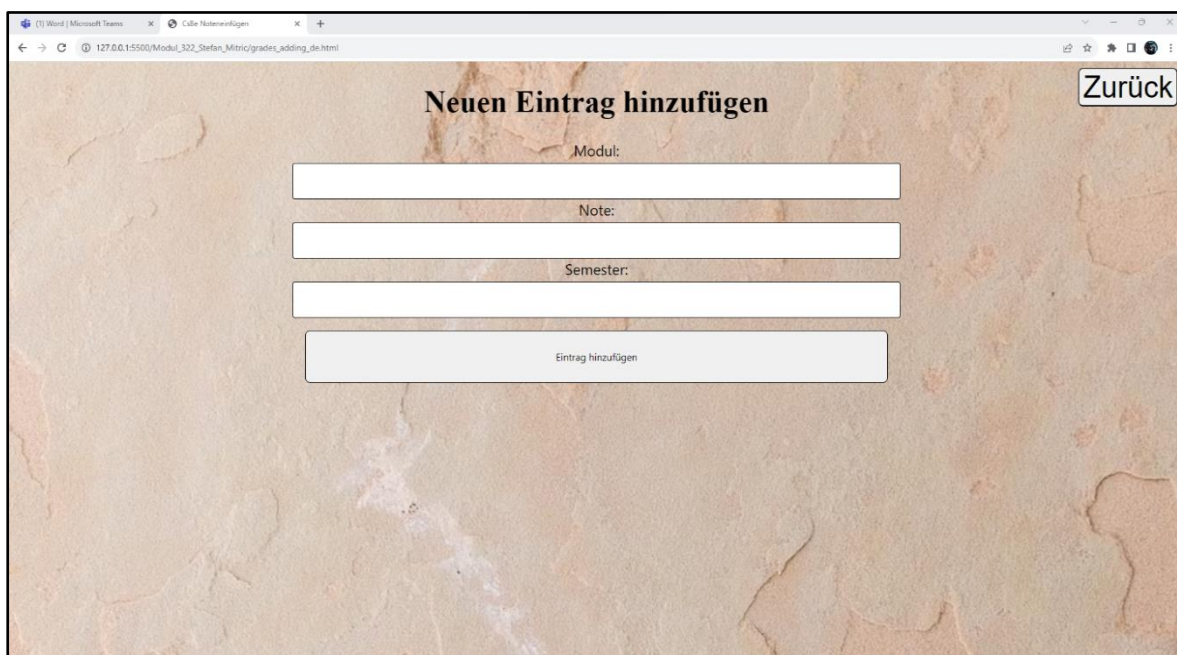


Abbildung 3: Noteneintragen

# Umsetzung

## Struktur

Die Struktur des Projekts wurde unter Verwendung von HTML, CSS und JavaScript implementiert. Jede dieser Technologien hat eine spezifische Rolle in der Entwicklung des Projekts gespielt.

### HTML

HTML (HyperText Markup Language) dient als Grundgerüst der Anwendung. Es wurde verwendet, um die grundlegenden Elemente und den Aufbau der Benutzeroberfläche zu definieren. Dazu gehören Formulare für die Eingabe von Daten, Buttons für Interaktionen und Tabellen zur Darstellung von Informationen.

### CSS

Das Styling der Anwendung wurde mit CSS (Cascading Style Sheets) realisiert. CSS wurde verwendet, um die visuellen Aspekte der HTML-Elemente zu gestalten. Dazu gehören Farben, Schriftarten, Layout und Responsivität. Mit Hilfe von CSS konnten wir ein benutzerfreundliches und ästhetisch ansprechendes Design schaffen, das die Benutzererfahrung verbessert.

### JavaScript

JavaScript wurde für die clientseitige Logik der Anwendung eingesetzt. Mit JavaScript wurde die Interaktivität der Benutzeroberfläche ermöglicht. Funktionen wie Formularüberprüfung, dynamisches Laden von Daten und die Implementierung von benutzerdefinierten Logiken wurden mit JavaScript realisiert.

## Testing

### Testkonzept

Sobald alle Features der Applikation umgesetzt wurden, wird erstmals die Testphase eingeleitet. Nach jeder Änderung, die nach der Testphase ausgeführt wird, wird eine neue Testphase gestartet und das Testprotokoll wird erneut durchgearbeitet.

Ein Release einer Version, für die das Testprotokoll nicht durchgearbeitet wurde, ist nicht gestattet.

### Testfälle

Dieser Abschnitt behandelt alle Testfälle, die im zwei Schritten der Testphase durchzuführen sind.

Als Vorbedingung für alle Testfälle gilt, dass das Programm gestartet wurde und das Hauptfenster angezeigt wird.



<b>Testfall 1: Kann man jede Sprache ansteuern?</b>
<b>Ablauf</b>  <ol style="list-style-type: none"><li>1. Im Login Menu alle Sprach-buttons drücken</li><li>2. Mit jeder Sprache die Notenübersicht ansteuern.</li><li>3. Mit jeder Sprache die Noteneingabe von der Notenübersicht ansteuern.</li></ol>



<b>Testfall 2: Login fehlschlagen</b>
<b>Ablauf</b>  <ol style="list-style-type: none"><li>1. Beim Login nichts eingeben.</li><li>2. Login drücken.</li></ol>



## Referenzen

GPT-4:  
Modul 293