

# Internetauftritt Studentenkasino

## Projektdokumentation

Studienjahrgang: 2024

Kurs: A

Fakultät: Wirtschaft

Studiengang: Wirtschaftsinformatik

Modul: Fallstudien / Herr Schulik

Semester: Sommersemester 2025

### Projektteam:

Corvin Annen – Projektleitung

Elijah Mossmann – Backend

Timo Manz – Frontend

Michael Zerner – Hosting & Deployment

DHBW Villingen-Schwenningen

Abgabedatum: 27.11.2025

# Inhaltsverzeichnis

Abkürzungsverzeichnis .....	III
Abbildungsverzeichnis .....	IV
1 Einleitung .....	1
1.1 Problemstellung .....	1
1.2 Zielsetzung .....	1
1.3 Aufbau .....	2
2 Projektmanagement .....	3
2.1 Projektmethodik .....	3
2.2 Austausch .....	3
2.3 Planung .....	4
2.3.1 Rollenverteilung .....	4
2.3.2 Zeitplan .....	5
2.3.3 Meilensteine .....	5
3 Konzeption .....	6
3.1 Anforderungen .....	6
3.2 Technologische Entscheidungen .....	6
3.3 Gesamtarchitektur .....	7
4 Backend .....	8
4.1 API-Design .....	8
4.2 Datenbankmodell .....	8
5 Frontend .....	8
5.1 Technologie-Stack .....	8
5.2 Komponentenstruktur .....	8
6 Hosting .....	8
6.1 Server-/Hosting-Modell .....	8
6.2 Deployment-Prozess .....	8
7 Projektergebnis/Reflexion .....	8
8 Anhang .....	8
9 Literaturverzeichnis .....	8

## **Abkürzungsverzeichnis**

## **Abbildungsverzeichnis**

# 1 Einleitung

## 1.1 Problemstellung

Die vorliegende Arbeit behandelt die Konzeption und Umsetzung einer Webanwendung, die sowohl eine gute Performance wie auch eine komfortable Skalierbarkeit enthalten soll. Moderne Webprojekte stehen vor der Herausforderung, gleichzeitig nutzerfreundlich, zuverlässig und bei wachsender Nutzung skalierbar zu bleiben. Besonders bei Anwendungen, die spielerische Elemente oder Wettbewerbsmechanismen enthalten, steigt die technische Komplexität deutlich, da Echtzeitinteraktionen und schnelle Datenverarbeitung erforderlich sind. Ein weiteres Problem ist die sichere Verwaltung von Benutzerdaten und Spielwährungen, da diese Systeme potenzielle Angriffsziele darstellen. Gleichzeitig muss die Oberfläche intuitiv gestaltet werden, um auch Studierende ohne technische Vorkenntnisse eine Spielerfahrung zu bieten. Die Plattform soll darüber hinaus flexibel erweiterbar sein, sodass neue Spiele, Module oder Funktionen ohne große strukturelle Umbauten integriert werden können. Zusätzlich wird im Rahmen der Arbeit untersucht, welche Marketingmethoden angewandt werden müssen um die gewünschte Zielgruppe zu erreichen. Insgesamt besteht die zentrale Herausforderung darin, eine robuste, skalierbare und benutzerorientierte Webanwendung zu entwickeln, die sowohl technisch als auch organisatorisch langfristig betreibbar ist.

## 1.2 Zielsetzung

Ziel der Arbeit ist die Entwicklung eines sicheren, skalierbaren und modular aufgebauten Studentencasinos, das durch spielerische Elemente den Wettbewerb zwischen den Nutzern fördert. Dazu wird eine fiktive Spielwährung eingeführt, der „Coin“, der als zentrale Ressource für alle Spiele und Mechaniken dient. Ein übergreifendes Punktesystem soll alle Spieler anhand ihrer Coin-Anzahl in einer Rangliste absteigend sortieren und so einen dauerhaften Anreiz zur Teilnahme schaffen. Durch verschiedene Minigames erhalten die Nutzer die Möglichkeit, ihre Coins zu vermehren, wodurch Dynamik und Variation entstehen. Die Plattform wird so entworfen, dass neue Spiele ohne größere strukturelle Anpassungen integriert werden können, was langfristige Erweiterbarkeit sicherstellt. Ein weiterer

Schwerpunkt liegt auf der Sicherheit des Systems, insbesondere der korrekten und manipulationssicheren Verwaltung der Spielwährung. Zusätzlich berücksichtigt die Zielsetzung Aspekte wie Nutzerfreundlichkeit, klare Interface-Gestaltung und die Schaffung eines motivierenden Spielerlebnisses. Insgesamt soll eine Plattform entstehen, die stabil, attraktiv und technisch sauber umgesetzt ist.

### **1.3 Aufbau**

Die Arbeit gliedert sich in mehrere Hauptkapitel, die den gesamten Projektverlauf von der Planung bis zur Reflexion strukturiert darstellen. Nach der Einleitung folgt das Kapitel Projektmanagement, in dem die gewählte Methodik, die Organisation des Teams sowie Kommunikations- und Planungsprozesse beschrieben werden. Anschließend widmet sich das Kapitel Konzeption den ermittelten Anforderungen, der technologischen Auswahl und der Ausarbeitung der Gesamtarchitektur des Systems. Die darauf folgenden Kapitel Backend und Frontend beschreiben im Detail die technische Umsetzung, die verwendeten Frameworks und die Art, wie Logik, Datenhaltung und Benutzeroberfläche miteinander interagieren. Im Kapitel Hosting wird erläutert, wie die Anwendung bereitgestellt, versioniert und deployt wird, um eine stabile und skalierbare Umgebung sicherzustellen. Danach folgt das Kapitel Projektergebnis/Reflexion, das die wesentlichen Resultate zusammenfasst und den Entwicklungsprozess hinsichtlich Effizienz, Herausforderungen und gewonnener Erfahrungen bewertet. Die Struktur der Arbeit ermöglicht eine klare Nachvollziehbarkeit der Entscheidungen sowie der technischen und organisatorischen Umsetzung des Projekts. Dadurch wird sowohl die fachliche Tiefe als auch die praktische Relevanz der Arbeit sichtbar gemacht.

## 2 Projektmanagement

In dem Kapitel Projektmanagement werden die organisatorischen und methodischen Grundlagen des Projekts beschrieben. Es umfasst die gewählte Projektmethodik, die Kommunikations- und Austauschprozesse innerhalb des Teams sowie die Planung des Projekts. Ziel dieses Kapitels ist es, einen klaren Überblick über die Strukturierung und Steuerung des Projekts zu geben, um die Effizienz und den Erfolg der Umsetzung sicherzustellen.

### 2.1 Projektmethodik

Für die Umsetzung des Projekts wurde eine iterative, agile Projektmethode mit regelmäßigen Meetings gewählt. Diese Methode ermöglicht es, flexibel auf Änderungen zu reagieren und kontinuierlich Verbesserungen vorzunehmen. Durch kurze Entwicklungszyklen konnten Anforderungen schnell umgesetzt und getestet werden, was die Qualität des Endprodukts erhöhte. Die agile Methodik förderte zudem die Zusammenarbeit im Team, da regelmäßige Abstimmungen und Feedbackrunden eingeplant wurden. Insgesamt trug diese Vorgehensweise dazu bei, das Projekt effizient zu steuern und den Fokus auf die wichtigsten Ziele zu legen.

### 2.2 Austausch

Die Kommunikation und der Austausch im Projektteam erfolgten überwiegend über eine gemeinsame WhatsApp-Gruppe sowie durch persönliche Gespräche im Anschluss an Vorlesungen. Für die kontinuierliche Abstimmung kamen OneNote und verschiedene Funktionen von GitHub zum Einsatz. GitHub ermöglichte es, Kommentare direkt im Code zu hinterlassen und Aufgaben eindeutig zuzuweisen. Zur Planung und Nachverfolgung einzelner Arbeitspakete wurde der „Issues“-Bereich von GitHub genutzt, der eine strukturierte Verwaltung von Milestones, Tasks und Fehlerberichten erlaubte. Auf dieser Grundlage konnten Fortschritte transparent dokumentiert und Prioritäten festgelegt werden. Ergänzend fanden wöchentliche Meetings über Discord statt, um komplexere Fragestellungen zu besprechen, Entscheidungen zu treffen und den direkten Austausch zu stärken. Zentrale Projektdokumente und relevante Informationen wurden in OneNote abgelegt, sodass alle Teammitglieder jederzeit Zugriff auf den aktuellen

Wissensstand hatten. Durch die kombinierte Nutzung dieser Werkzeuge wurde eine transparente und effiziente Kommunikation ermöglicht, was wesentlich zur erfolgreichen Zusammenarbeit im Projekt beitrug.

## 2.3 Planung

Die Planung bildete den organisatorischen Rahmen des Projekts und strukturierte die Zusammenarbeit über den gesamten vierwöchigen Bearbeitungszeitraum. Zu Beginn wurden Zuständigkeiten definiert sowie Anforderungen und Ziele des geplanten Internetauftritts festgelegt. Darauf aufbauend entstand eine Projektstruktur, aus der sich Aufgaben, Meilensteine und zeitliche Prioritäten ableiten ließen. Die anschließende Feinplanung diente als Grundlage für die Ausarbeitung der Rollen, die Erstellung eines realistischen Zeitplans sowie die Festlegung zentraler Meilensteine. Während der Umsetzung wurde diese Planung kontinuierlich überprüft und bei Bedarf angepasst, um den engen Zeitrahmen effizient zu nutzen.

### 2.3.1 Rollenverteilung

Die Rollenverteilung im Projektteam wurde klar definiert, um eine effiziente Zusammenarbeit und Verantwortungsübernahme zu gewährleisten. Corvin Annen übernahm die Projektleitung und koordinierte sämtliche organisatorischen Abläufe. Dazu gehörten Terminabsprachen, die Abstimmung zwischen den Entwicklungsbereichen sowie die Überwachung des Projektfortschritts. Elijah Mossmann war für die Backend-Entwicklung verantwortlich. Sein Aufgabenbereich umfasste die Implementierung der API-Strukturen, die Anbindung der Datenbank sowie die Sicherstellung der funktionalen Logik. Timo Manz leitete die Frontend-Entwicklung. Er konzipierte die Benutzeroberfläche, setzte UI-Komponenten um und sorgte dafür, dass Funktionalität und Gestaltung konsistent ineinandergreifen. Michael Zerner betreute Hosting und Deployment. Er kümmerte sich um die Serverumgebung, richtete das Hosting ein und verantwortete die Bereitstellung der Anwendung einschließlich Tests der Live-Version. Diese Rollenverteilung gewährleistete, dass Arbeitsbereiche klar abgegrenzt waren und jedes Mitglied zielgerichtet beitragen konnte.

### **2.3.2 Zeitplan**

Für das vierwöchige Projekt wurde ein komprimierter, aber realistischer Zeitplan ausgearbeitet. Die erste Woche war der Konzeption gewidmet: Klärung der Anforderungen, Auswahl geeigneter Technologien und Festlegung der Systemarchitektur. In den beiden folgenden Wochen erfolgte die parallele Entwicklung von Backend und Frontend, begleitet vom Aufbau der Datenbank und der Implementierung grundlegender UI-Elemente. In der letzten Woche standen Hosting, Integrationstests und Optimierungen im Vordergrund. Regelmäßige Abstimmungen erlaubten es, Fortschritte zu kontrollieren und den Zeitplan situativ anzupassen, ohne die Zieltermine zu gefährden.

### **2.3.3 Meilensteine**

Die Meilensteine des Projekts wurden strategisch festgelegt, um den Fortschritt zu überwachen und sicherzustellen, dass die wichtigsten Ziele erreicht werden. Zu den zentralen Meilensteinen gehörten die Fertigstellung der Konzeptionsphase, die Implementierung des Backends und Frontends, die Entwicklung der Datenbankstruktur sowie die Bereitstellung der Anwendung im Hosting-Umfeld. Jeder Meilenstein markierte einen bedeutenden Fortschritt im Projektverlauf und diente als Grundlage für die nächsten Schritte. Durch die klare Definition dieser Meilensteine konnte das Team den Überblick behalten und sicherstellen, dass alle Aufgaben termingerecht abgeschlossen wurden.

## 3 Konzeption

Das Kapitel Konzeption beschreibt die grundlegenden Überlegungen und Entscheidungen, die der technischen Umsetzung des Projekts zugrunde liegen. Es umfasst die Analyse der Anforderungen, die Auswahl der technologischen Komponenten sowie die Ausarbeitung der Gesamtarchitektur des Systems. Ziel dieses Kapitels ist es, eine klare und nachvollziehbare Grundlage für die anschließende Entwicklung zu schaffen.

### 3.1 Anforderungen

Die Anforderungen an den Internetauftritt des Studentencasinos wurden sorgfältig analysiert, um eine benutzerfreundliche, sichere und skalierbare Plattform zu schaffen. Funktionale Anforderungen umfassen die Implementierung eines Punktesystems mit einer fiktiven Spielwährung („Coins“), die Integration verschiedener Minigames zur Coin-Generierung sowie die Anzeige einer Rangliste basierend auf der Coin-Anzahl der Nutzer. Darüber hinaus soll die Plattform modular aufgebaut sein, um zukünftige Erweiterungen und neue Spiele problemlos integrieren zu können. Nicht-funktionale Anforderungen beinhalten Aspekte wie Sicherheit bei der Verwaltung der Spielwährung, Performance-Optimierung für eine reibungslose Nutzererfahrung sowie eine intuitive Benutzeroberfläche. Zusätzlich wurde Wert auf eine klare und ansprechende Gestaltung gelegt, um die Zielgruppe der Studierenden effektiv anzusprechen. Insgesamt sollen diese Anforderungen sicherstellen, dass die Plattform sowohl technisch robust als auch attraktiv für die Nutzer ist.

### 3.2 Technologische Entscheidungen

Die technologischen Entscheidungen für die Umsetzung des Studentencasinos wurden auf Basis der Anforderungen und Zielsetzungen des Projekts getroffen. Für das Backend wurde das Django-Framework in Python gewählt, da es eine robuste Struktur für die Entwicklung von Webanwendungen bietet und eine einfache Integration mit Datenbanken ermöglicht. Django's integriertes Authentifizierungssystem erleichtert zudem die sichere Verwaltung von Benutzerdaten.<sup>1</sup> Für das

---

<sup>1</sup>Dhadil u. a., „Django Unleashed: A Deep Dive into the Features and Advantages of the Django Framework“.

Frontend fiel die Wahl auf React, da es eine komponentenbasierte Architektur bietet, die eine flexible und wiederverwendbare Gestaltung der Benutzeroberfläche ermöglicht. Vite wurde als Bundler eingesetzt, um schnelle Entwicklungszyklen und effizientes Build-Management zu gewährleisten. Als Datenbank wurde SQLite verwendet, da es leichtgewichtig ist und sich gut für die Anforderungen des Projekts eignet, mit der Möglichkeit, bei Bedarf auf leistungsfähigere Systeme wie MySQL oder PostgreSQL zu skalieren. Diese technologischen Entscheidungen tragen dazu bei, eine stabile, skalierbare und benutzerfreundliche Plattform zu schaffen.

### 3.3 Gesamtarchitektur

#### Backend (API)

- Framework: Django (Python)
- Sprache: Python
- API Endpoints:
  - GET /api/users
  - POST /api/users
  - GET /api/posts
- Authentifizierung: Django Auth System
- Geschäftslogik: Validierung und Autorisierung durch Django-Modelle und Middleware
- Datenbank: SQLite (mit Django ORM)

#### Frontend (Client)

- Framework: React (mit Vite als Bundler)
- Sprache: JavaScript
- Wichtige Komponenten:
  - UI-Komponenten: React-Komponenten, Seiten
  - Datenabfrage: Client-seitiges Fetching mit AJAX
  - Navigation: React Router für client-seitige Navigation

#### Datenbank

- Datenbank: SQLite (kann für Produktion auf MySQL/PostgreSQL erweitert werden)
- Protokoll: SQL

- Datenbank-Komponenten: Tabellen, Indizes, Constraints, verwaltet durch Django ORM

UI-Komponenten

- Sprache: HTML, CSS, JavaScript
- Framework: React
- Server-Side Rendering: Optional, mit Next.js für SSR (nicht im Repo explizit genutzt)

## 4 Backend

### 4.1 API-Design

### 4.2 Datenbankmodell

## 5 Frontend

### 5.1 Technologie-Stack

### 5.2 Komponentenstruktur

## 6 Hosting

### 6.1 Server-/Hosting-Modell

### 6.2 Deployment-Prozess

## 7 Projektergebnis/Reflexion

## 8 Anhang

## 9 Literaturverzeichnis

[1] S. Dhadil u. a., „Django Unleashed: A Deep Dive into the Features and Advantages of the Django Framework“, *ResearchGate preprint*, 2024, [On-

line]. Verfügbar unter: [https://www.researchgate.net/publication/384557744\\_Django\\_Unleashed\\_A\\_Deep\\_Dive\\_into\\_the\\_Features\\_and\\_Advantages\\_of\\_the\\_Django\\_Framework](https://www.researchgate.net/publication/384557744_Django_Unleashed_A_Deep_Dive_into_the_Features_and_Advantages_of_the_Django_Framework)