Informationen zur Prüfungsleistung

In diesem Dokument stelle ich euch alle Infos bereit, die ihr benötigt, um euch auf die Prüfung vorzubereiten.

Zusammenfassung

Ihr werdet eine kleine Full-Stack-App entwickeln, in der ihr euch um die gesamte Spannbreite vom Frontend bis zur Datenbank kümmert.

Es geht nicht um ein perfektes Resultat, sondern um das Erlernen aller Schritte, die ihr benötigt, um auch nach dieser Prüfung immer wieder schnell Anwendungen bauen zu können.

Arbeitsumfang

Die gesamte Projektarbeit soll im Rahmen der 30 Vorlesungseinheiten sowie knapp 20 Stunden zuhause erledigt werden.

Anforderungen an das Projekt

Sucht euch ein Projekt aus, das folgende Anforderungen erfüllt

Pflichtanforderungen

- Ein echtes Problem wird gelöst
    - “echt” bedeutet:
        - Es gibt Menschen da draußen, die dieses Problem haben
        - Eure Lösung funktioniert nicht wie die 1000 anderen, die vielleicht schon existieren.
- Es gibt einen Bedarf für ein Web-basiertes Frontend
    - Keine Apps, die vollständig automatisiert laufen, sondern auf Eingaben von Nutzern basieren etc.
- Es gibt einen Bedarf für serverseitige Logik
- Die Lösung bedarf einer (minimalen) Form von Persistenz in einer Datenbank
- Als Dozent kann ich das Projekt in meinem Dev-Environment ausführen und testen. Nutzt also keine exotischen Dependencies, die nicht frei verfügbar sind.

Optionale Anforderungen

- Authentifizierung: Falls euer Produkt zwingend mehrere Benutzer erfordert, um zu funktionieren, solltet ihr das auch implementieren.
    - Nutzt dafür **better-auth** oder eine andere Lösung eurer Wahl. Wenn ihr Auth selber implementiert, seid euch sicher, dass ihr wisst, was ihr tut. Halbgarer Eigenbau gibt Punktabzug!

Sonstige Anforderungen

- Solltet ihr weitere Abhängigkeiten nutzen, stellt sicher, dass andere (wie z.B. ich als Dozent) diese Abhängigkeit ebenfalls nutzen können.

Bewertungskriterien

- Euer Projekt entspricht einem MVP der Idee.
    - Heißt: Der wichtigste Teil der Nutzererfahrung, der einen Mehrwert schafft, wurde umgesetzt.
- Erfüllung aller Pflichtanforderungen
- Der Code ist verständlich und sauber geschrieben. Heißt:
    - Es gibt keine signifikanten Anteile an unused code
    - Komplexere Abschnitte werden kommentiert
- Euer Produkt ist einfach zu nutzen. Alle Funktionen sind klar erklärt und erreichbar.
    - Das könnt ihr super prüfen, indem ihr eure Projekte durch Kommilitonen testen lässt.

Ideenfindung

Um eine Idee für die Projektarbeit zu finden, könnt ihr einen der folgenden zwei Ansätze wählen.

Variante 1: Ihr findet eine Lösung für ein Problem aus eurem Umfeld

Sucht euch ein mit digitalen Produkten lösbares Problem aus eurem Umfeld. Denkt an Probleme von:

- Eurer Familie (Eltern, Geschwister)
    - z.B. “Wir brauchen einen simplen digitalen Familienkalender”
- Eurem Partnerunternehmen
    - z.B. “Die Marketingabteilung benötigt eine einfachere Möglichkeit, Kundendaten für Kampagnen zu verwalten"
- Euch selbst

- z.B. “Ich muss mir sehr viele Fachbegriffe merken und das ist schwierig.”

Und zur Not fragt ihr KI 🙂

Anschließend überlegt ihr euch, wie ihr das Problem schnell und einfach lösen könnt. Bspw. könnte eine Karteikarten-App für das Problem der vielen Fachbegriffe eine gute Lösung sein. Vielleicht bringt ihr einen interessanten Twist rein (gamification?), der das ganze dann spaßiger macht.

Variante 2: Eine Lösung für ein vorgegebenes Problem

Falls ihr keine interessanten Probleme aus eurem Umfeld kennt, die ihr für lösbar hält, könnt ihr euch gerne an dieser Liste an Problemen bedienen:

1. **WG-Chaos**
   In vielen Wohngemeinschaften (WGs) sind die Absprachen für Putzpläne, Einkaufslisten und die Abrechnung gemeinsamer Ausgaben unorganisiert und führen oft zu Diskussionen. Kann man hier eine einfache Lösung bauen, die Kosten und Aufgabenteilung managed?

2. **Verlorengegangenes Wissen in Unternehmen**
   Gerade zu Beginn einer Praxisphase oder bei neuen Projekten werden Studierende mit unzähligen neuen Abkürzungen, Prozessen und Ansprechpartnern konfrontiert, die nirgends zentral dokumentiert sind. Vielleicht schafft hier eine einfach zu nutzende Wissensbasis abhilfe?

3. **Skill-Tauschbörse**
   Viele Studierende besitzen Fähigkeiten, die sie gerne weitergeben würden, oder suchen nach Unterstützung in einem bestimmten Bereich, wissen aber nicht, wer ihnen helfen könnte. Vielleicht kann hier eine dedizierte Plattform abhilfe schaffen?

4. **Schlaue Notizensammlung**
   Beim Recherchieren im Internet stößt du ständig auf interessante Artikel, Tools oder Videos, hast aber keine gute Methode, diese thematisch zu sortieren und mit eigenen Notizen für später zu speichern. Browser-Lesezeichen sind unübersichtlich und erlauben keine einfache Kategorisierung oder das Hinzufügen von Kontext.

5. **Die ultimative Studenten To-Do-List**
   Wie kannst du den Überblick über die vielen verschiedenen Aufgaben, Abgabetermine und Lernphasen für deine Uni-Module oder private Projekte behalten, wenn klassische To-do-Listen zu unstrukturiert sind? Es fehlt oft eine

zentrale Ansicht, die nicht nur auflistet, was zu tun ist, sondern auch dabei hilft, den Fortschritt über die Zeit zu visualisieren und Prioritäten zu setzen.

“Aber es gibt schon so viele Lösungen”

Gut! Bei acht Milliarden Menschen auf dieser Erde ist zu erwarten, dass es für Probleme bereits gute Lösungen gibt, die vielleicht sogar ähnlich wie eure Idee funktionieren.
Betrachtet das als Validierung dafür, dass ihr euch einem echten Problem annimmt (und das ist gut, denn sonst würde niemand euer Produkt kaufen).

Wie könnt ihr eure Lösung also besonders machen? Ganz einfach: Versucht, einen Aspekt des Projekts auf neuartige und interessante Weise zu lösen.

Ein Beispiel: Eine Karteikarten-App, bei der die Zeit abläuft und bei richtiger Auflösung zusätzliche Lebenszeit gegeben wird, macht das Lernen weniger langweilig.

Tech-Stack

Ich überlasse euch die komplett freie Wahl bzgl. des gewählten Tech-Stacks.

Aber: Die Vorlesung wird sich um einen gewissen Stack drehen, der so bzw. in ähnlicher Form aktuell sehr populär ist und für die meisten Cases gut funktioniert. Zudem werde ich vor dem Beginn des zweiten Vorlesungsblocks noch ein Basis-Repository bereitstellen, mit dem ihr dann arbeiten könnt.

Wenn ihr also den “empfohlenen” Weg geht, könnte es das für euch einfacher machen. Wenn ihr aber schon Fan von einem anderen Stack seid, nutzt diesen bitte.

Wichtig: Wer keinen Standard-Stack nutzt, sollte sicherstellen, dass ich die von euch entwickelte App tatsächlich trotzdem selbst ausführen kann. Also bitte bitte keine komischen DB-Anbieter oder Stacks, die ich nicht einfach selbst zusammenstecken kann. Danke!

Referenz Tech-Stack

Meta-Framework: Next.js 15
Frontend-Framework: React 19
Datenbank ORM: Drizzle
Datenbank: SQLite (ist das einfachste für diese Vorlesung, in Prod nutzt man natürlich wiederum andere Dinge wie PostgreSQL)

Authentifizierung: better-auth
Styling: TailwindCSS 4
Komponenten: shadcn/ui

Zu diesem Tech-Stack findet ihr einen riesigen Haufen an Infos online und jedes populäre LLM hat diesen Stack mit am besten verinnerlicht. Weitere Infos und Links etc. findet ihr zudem im Vorlesungsmaterial.

Informationen zur Einrichtung des Referenz-Repositories (das könnt ihr dann als Basis verwenden) findet ihr in der README vom Repository. Der Link dazu wird noch im Moodle-Kursraum geteilt.

Das Referenz-Repository wird euch auch einen generischen Minimalaufbau mit typischen Layouts mitliefern, die ihr nutzen könnt, aber nicht müsst!

Zudem wird das Repository eine AGENTS.md mitliefern, mit der viele Agentic Coding Tools klarkommen. Zur Not könnt ihr diese Datei auch als Kontext in einen externen LLM-Chat reingeben.