bewertungs raster

Für die Lehrperson: 223-07A - 02.00 Beispielfragen Fachgespräch (Web view)

Bewertungs raster	**Max-Punkte**
Dokumentation sinnvoll strukturiert und durchgängig einheitlich formatiert	**2**
drei User Stories mit Akzeptanzkriterien aussagekraftig, korrekt und mit eigenen Worten dokumentiert (in GitHub)	**3**
Backend funktional und korrekt implementiert und dokumentiert	**3**
mind. 2 sinnvolle Backend-Tests automatisiert und protokolliert	**2**
Backend-Technologien und -Architektur aussagekraftig, korrekt und mit eigenen Worten beschrieben und illustriert (inkl. sinnvoll eingesetzter Transaktionen)	**3**
Frontend funktional und korrekt implementiert und dokumentiert	**3**
mind. 2 Frontend-Tests automatisiert und protokolliert	**2**
Frontend-Technologie und -Architektur aussagekraftig, korrekt und mit eigenen Worten beschrieben und illustriert	**2**
Git korrekt eingesetzt (mit Branches, mehreren ordentlich kommentierte Commits)	**5**
JWT-Authentifizierung im Front- und Backend korrekt implementiert und im Code dokumentiert	**3**
Sicherheitskonzept dokumentiert	**2**
Live Produktpräsentation (max. 10 Minuten)	**5**
Fachgespräch / Kurztest	**10**

1. Titelblatt (Nicht nötig, siehe Readme von Superhero App)
2. Einleitung und Anforderungsanalyse (User Stories)
3. Sicherheitskonzept
4. Arbeitsplanung (sinnvolle Arbeitspakete mit geschätztem Zeitaufwand) (Nicht nötig)
5. Test-Konzept (Nich nötig)
6. Kurze Beschreibung der eingesetzten Frameworks
7. Kurze Beschreibung der Abläufe beim Login
8. Testprotokoll (2 FE / 2 BE Test und Bilder der Resultate in der Doku)
9. Arbeitsjournal (1 Eintrag pro Block)

Sicherheitskonzept

1. Authentifizierung & Autorisierung
   - JWT-Token Implementation
   - Rollenbasierte Zugriffskontrolle (ADMIN/USER)
   - Sichere Token-Speicherung
   - Token-Invalidierung beim Logout
   - Weiterleitung bei abgelaufenen Tokens

2. Datensicherheit
   - Passwort-Hashing und Salting
   - HTTPS/TLS-Verschlüsselung
   - Input-Validierung
   - Keine sensiblen Daten im Frontend-Code

3. API-Sicherheit
   - Grundlegende Rate-Limiting Implementation
   - Secure Headers (CORS, CSP)
   - Zugriffsschutz für API-Endpunkte
   - Validierung der API-Anfragen

4. Frontend-Sicherheit
   - XSS-Schutz durch React
   - Geschützte Routen (ProtectedRoute)
   - Validierung von Benutzereingaben
   - Sichere State-Verwaltung

Arbeitsjournal

Format:
Datum: XX.XX.2024
Zeit: 14:00-17:00
Person(en): Name(n)
Tätigkeiten:
- Durchgeführte Arbeiten
- Aufgetretene Probleme

Beispiel:

Datum: 15.01.2024
Zeit: 09:00-12:00
Person: Anna
Tätigkeiten:
- JWT Authentication im Backend implementiert

Person(en): Name(n)
Tätigkeiten:
- Durchgeführte Arbeiten
- Aufgetretene Probleme
- Lösungen
Status: Erledigt/In Bearbeitung
Nächste Schritte:
- Geplante Tasks

Zeit: 09:00-12:00
Person: Anna
Tätigkeiten:
- JWT Authentication im Backend implementiert
- User Entity erstellt
- Probleme mit Token-Validierung gelöst durch Debug-Session
Status: Erledigt
Nächste Schritte:
- Protected Routes im Frontend

Datum: 15.01.2024
Zeit: 13:00-16:00
Personen: Max, Tom
Tätigkeiten:
- SuperheroAPI Integration
- Frontend Battle-Component entwickelt
- Problem: API Rate Limiting
- Lösung: Caching implementiert
Status: In Bearbeitung
Nächste Schritte:
- Battle-Logik vervollständigen
- Unit Tests schreiben