

Programmieren 3 (Sommersemester 2024)

Portfolio-Prüfung

Hintergrund

Die Prüfungsleistung in der Vorlesung Programmieren 3 wird durch ein Portfolio erbracht. Hierbei sollen Sie zeigen, dass Sie die Inhalte der Vorlesung der zugehörigen Übungen verinnerlicht haben und praktisch anwenden können. Es wird darüber hinaus ausdrücklich von Ihnen erwartet, dass Sie sich weitere Inhalte *eigenständig* aneignen, die nicht in der Vorlesung und den Übungen vermittelt wurden!

Die Bearbeitung findet in Einzelarbeit statt.

Antritt

Die Anmeldung erfolgt via Campusportal sowie durch Ausgabe der Arbeit. Sofern Sie die Arbeit annehmen, gilt dies als Anmeldung – auch wenn Sie sich später im Campusportal wieder abmelden sollten.

Versionierung

Sämtliche während des Portfolios erbrachten Leistungen an Artefakten (Dokumente, Protokolle, Entwürfe, Code, Skizzen, Testfälle, Testpläne...) sind von den Bearbeitern in einem eigenen Git-Repository einzupflegen, damit die individuellen Leistungen nachvollziehbar und zuordenbar sind. Sie können dazu entweder ein lokales Repository nutzen oder ein entsprechendes Hosting-Angebot von Dritten in Anspruch nehmen.

Abgaben und Zwischenstände

Die Abgabe erfolgt per E-Mail.

Abnahme

Am Ende des Projekts erfolgt eine Abnahme in Form einer mündlichen Prüfung. Sie werden dabei zu Einzelheiten des von Ihnen erstellten Quellcodes befragt. Ziel hierbei ist es zu prüfen, ob Sie die von Ihnen eingereichten Beiträge tatsächlich in Eigenleistung erbracht haben. Sofern dies der Fall ist, haben Sie bei der Abnahmeprüfung nichts zu befürchten.

Aufgaben und Bewertung

Das Portfolio besteht aus mehreren zu leistenden Aufgaben, aus denen die folgenden Aktivitäten bzw. Abgaben resultieren:

- Vereinfachte Spezifikation (5%)
- Implementierung Progressive WebApp (55%)
- Implementierung Backup-Server (25%)
- Unit-Tests, Testplan, Testprotokoll (10%)
- Demonstrationsvideo (5%)

Der Wert in Klammern gibt an, zu wieviel Prozent die jeweilige Leistung in die Gesamtbewertung einfließt. Die einzelnen Bewertungskriterien der jeweiligen Leistungen ergeben sich aus den in der Vorlesung vorgestellten Qualitätskriterien für die einzelnen Aktivitäten und werden im Folgenden teilweise zusätzlich aufgeführt.

Szenario und Anforderungen

Die Aufgabe besteht darin, eine Progressive WebApp (PWA) für eine generische FAQ (Frequently Asked Questions) zu erstellen. Die Anwendung soll folgende funktionale Anforderungen erfüllen:

- Anzeige von FAQ-Einträgen (bestehend aus Kategorie, Frage und Antwort)
- Pflege von FAQ-Einträgen und Kategorien (die Anwendung soll bereits mit einem Datensatz ausgeliefert werden, der sich aber über ein verstecktes Menü erweitern lässt)
- Import und Export der Daten über eine gesicherte Web-API an einen Backup-Server

Optionale Anforderungen (Umsetzung von mindestens zwei davon notwendig, um die Note 1,7 oder besser zu erreichen):

- Darkmode
- Suchfunktion
- Filterfunktion
- Pagination
- Mehrsprachenfähigkeit
- Rich-Text für die Gestaltung der Texte in der Markdown-Syntax

Fragen zu den Anforderungen können per E-Mail an den Prüfer gestellt werden.

Aufgaben

(1) Erstellen Sie für die in diesem Dokument beschriebenen Anforderungen eine vereinfachte Spezifikation (mind. Datenmodell, Mengengerüst, UI-Skizzen - keine Use-Case-Beschreibungen).

(2) Implementieren Sie sowohl das spezifizierte Frontend mit HTML/CSS/JavaScript als PWA und den Backup-Dienst in Rust.

(3) Erstellen Sie Unit-Tests. Erstellen Sie zudem einen geeigneten Testplan mit einer Testsequenz für einen Smoketest, der maximal fünf Minuten Testaufwand in Anspruch nehmen darf. Machen Sie mindestens eine protokollierte Durchführung dieser Testsequenz, um die prinzipielle Funktionsfähigkeit Ihres Systems zu demonstrieren.

Verwendete Technologien

Die Hauptanwendung ist zwingend als PWA mit lokaler Datenhaltung zu realisieren. Hierfür dürfen nur ausdrücklich genehmigte Bibliotheken und Frameworks verwendet werden – Sie müssen die Anwendung daher vorwiegend mit „vanilla-JavaScript“ realisieren. Dies mag umständlich erscheinen, ist jedoch didaktisch ausdrücklich gewollt!

Der Backup-Dienst ist zwingend in Rust zu realisieren und muss über eine Web-API angesprochen werden. Für den Backup-Dienst dürfen beliebige Drittanbieter-Bibliotheken genutzt werden, die über crates.io verfügbar sind. Das Austauschverfahren zwischen der PWA und dem Backup-Dienst kann frei gestaltet werden. Sicherheitsanforderungen können vernachlässigt werden.

Geforderte Artefakte

Für die finale Abgabe muss folgende Artefakte enthalten:

- Vereinfachte Spezifikation (PDF)
- Gesamtes git-Repository samt Quellcode der implementierten Anwendung (Frontend+Backend) als ZIP-Archiv
- Testplan und Testprotokoll (PDF)
- Video (Bildschirmaufzeichnung) der Durchführung der Testsequenz (max. 3 Minuten), erlaubte Formate → siehe Moodle

Bewertungskriterien

Grundsätzlich sollten Sie alle in der Vorlesung erlernten Best Practices für die Nutzung der behandelten Technologien befolgen. Bei der Bewertung Ihrer Leistung werden unter anderem folgende Kriterien berücksichtigt:

- Allgemeine Spezifikationskriterien wie in „Software Engineering I“ behandelt
- Implementierter Umfang (es gilt jedoch das Prinzip „Klasse statt Masse“)
- Gestaltung und Bedienbarkeit der Oberfläche inkl. Responsivität (Nutzung kleinerer und größerer Bildschirme, Beachtung Darkmode, ...)
- Lesbarkeit und Strukturiertheit des Quellcodes
- Qualität von UNIT-Tests, Testdaten und Testplänen
- Qualität des Videos sowie der sonstigen Dokumentation

Grundsätzlich sind die Unter Aspekte der Produktqualität - wie in der Vorlesung „Software Engineering I“ behandelt - zu berücksichtigen.

Fristen

Abgabe Endfassung
Projektabschluss

05.07.2024, 18:00 Uhr

Im Prüfungszeitraum Sommersemester 2024 (d.h. im Juli), der Termin wird mit dem Prüfungsplan bekannt gegeben