Testbericht

Simple Chat (API & Frontend)

Inhalt

Grundinformationen:

Produktinformationen

Bei dem zu testenden Produkt handelt es sich um eine Kombination aus zwei getrennten, jedoch zusammenarbeitenden Programmen. Einerseits der im Backend arbeitende Webserver, welcher mittels des Python-Frameworks Bottle erstellt wurden und zum anderen ein in VueJS3 erstellte Frontend Single Page, reaktive Web-App welche die Chats entsprechend visualisiert und dem Nutzer Interaktionen mit dem Backend erlaubt d.h. das Bedienen erlaubt. Die Kommunikation zwischen den beiden Projektteilen verläuft über eine REST API welche Daten im JSON Format erwartet und ausliefert.

Testrahmen

In den Testvorgarben wurden keine spezifischen Testmethoden oder Mittel zum Testen beschrieben, jedoch sollte eine an die Verhältnisse des Programms angepasste, ausreichende, Testung durchgeführt werden. Um dies zu gewährleisten wurde sich für Folgende Aufteilung entschieden:
Das gesamte Backend, vornehmlich allerdings die API Endpointes sind Maschinell, über Unittests auf ihre Funktionalität zu überprüfen. So kann sichergestellt werden, das nach jeder Änderung auch Edge-cases Testabdeckung findet. Des weiteren liegt sowohl bei den Datenbankfunktionen als auch, durch Verwendung einer REST API eine klare Datenstruktur vor, die sich als Grundlage für maschinelle Tests eignet. Auch wären händische Tests, gerade langfristig gesehen Aufwendiger, da nicht bloß die Korrektheit der UI, überprüft werden muss.
Im Frontend hingegen sind maschinelle Tests etwas schwerer zu realisieren und gleichzeitig sind auftretende Fehler, angenommen das Backend arbeitet korrekt weniger gravierend, bzw. möglicherweise nur einzelne Geräte oder Nutzer betreffend, die Datenintegrität der Datenbank ist nicht gefährdet. Deshalb wurde sich hier für händischen Testung, durchgeführt von verschiedenen Personen entschieden. Dies bat außerdem die Möglichkeit die Intuitivität der Nutzeroberfläche zu erproben.

API & Server Tests

Alle erstellten Tests sind im Verzeichnens `./simple_chat_api/tests` einsehbar. Diese sprechen grundsätzlich für sich selbst. Es wurde auf folgendes geachtet:

- Jeden Fall mit Nutzern aller Berechtigungsstufen zu testen
- Auch auf Fehler zu testen/diese zu erwarten
- Die erwarteten Rückgabewerte im Testtulpe so eng wir möglich zu gestalten
- Die Abhängigkeiten von Testfunktionen aufeinander klar zu markieren, die Unabhängigkeit der Tests jedoch so weit wie möglich sich zustellen
- Jeden Existierenden API-Endpoint mit mindestens einem Test abzudecken

Zum Zeitpunkt der Erstveröffentlichung waren alle Tests Erfolgreich und Fehlerfrei ausgeführt. Jedoch muss angemerkt werden das die Skalierbarkeit der Tests durch das trennen in verschiedene Test Suits gesteigert werden kann und bei wachsender Funktionalität des Produktes muss. Auch können perspektivisch die existierenden Tests um weitere Tuple vergrößert werden um ggf. noch unbekannte Edgecases abzudecken. Jedoch würde ich abschließend schlussfolgern das zum jetzigen Entwicklungsstandpunkt, bedenkt man den Umfang des Projektes eine ausreichende Testung vorliegt und durch ihre Fehlerfreiheit die Funktionalität des Servers gewährleistet werden kann.

Frontend Tests

Für das Frontend wurden Händische Tests vorgenommen, diese konnten keine gravierenden Fehler feststellen. Folgende Anmerkungen sind jedoch hervorzuheben:

- Admin Panel, Mobile/Small Viewport: Content Boxen sind nicht zentriert
- Nach ablaufen des Coockies der den Nutzer angemeldet hält, wird er Abgemeldet, es existiert jedoch kein Info--fenster das Informiert warum

Bei beiden Punkten handelte es sich um Fehlende Funktionalität oder Designtechnische Fehler, die nicht Funktionsbrechend sind. Somit kann auch hier die Funktionalität, in Verhältnismäßigkeit zum erwarteten Anwendungszweck festgestellt werden.

Fazit

Beide Tests für Front/-Backend konnten die erwartungsgemäße Funktionalität feststellen. Anzumerken ist jedoch das keine der gewählten Testmethoden eine 100% Sicherheit auf Fehlerfreiheit bietet. Besonders die Händischen Tests im Frontend sind kritisch zu betrachten.