

Studienjahr: SS 2024

Semester: 4

Projektteam

Scrum Master

<Chedjou, Ange, pauleange.chedjoumatsingang@edu.fh-kaernten.ac.at >

Entwicklungsteam

Frontend-Entwickler: <Knaus, Anastasiia, anastasiia.knaus@edu.fh-kaernten.ac.at >

Backend-Entwickler: <Steshkov, Vadim, vadim.steshkov@edu.fh-kaernten.ac.at>

Tester: <Zamanova, Nina, nina.zamanova@edu.fh-kaernten.ac.at >

Titel des Projekts

VogueVibes

Problembeschreibung

Das Bekleidungsgeschäft kämpft mit einer ineffizienten Bestandsverwaltung. Die genaue Überwachung der Lagerbestände gestaltet sich als Herausforderung, was dazu führt, dass bestimmte Artikel entweder in zu großer Menge oder zu knapp vorhanden sind. Dies hat direkte Auswirkungen auf die Umsätze, führt zu Platzverschwendung im Lager und einer ineffizienten Nutzung der vorhandenen Ressourcen. Zudem gestaltet sich die Vorhersage des Absatzes einzelner Artikel als äußerst schwierig, was das Problem der Bestandsverwaltung zusätzlich erschwert. Dieser Mangel an Vorhersagbarkeit führt dazu, dass das Geschäft Geld und Kunden verliert, da bestimmte Artikel entweder nicht verfügbar sind oder veraltet im Lager liegen. Eine dringende Lösung ist erforderlich, um die Bestände effektiver zu verwalten und die Vorhersagegenauigkeit zu verbessern. Nur so können fundierte Entscheidungen darüber getroffen werden, welche Artikel in welchen Mengen vorgehalten werden sollten, um den Bedürfnissen der Kunden gerecht zu werden.

Lösungsbeschreibung

Die vorgeschlagene Software bietet Bestandsaktualisierungen in Echtzeit, was es dem Bekleidungsgeschäft ermöglicht, seine Lagerbestände präzise und zeitnah zu verfolgen. Dadurch kann das Unternehmen schnell fundierte Entscheidungen über die Auffüllung oder Nachbestellung von Artikeln treffen. Zusätzlich liefert die Software wertvolle Einblicke in das Kaufverhalten der Kunden, was dem Geschäft dabei hilft, die Beliebtheit einzelner Artikel zu erkennen. Diese Erkenntnisse dienen als Grundlage für die Anpassung der Einkaufs- und Lagerhaltungsstrategien, um das Risiko von Über- oder Unterbevorratung zu minimieren.

Ein weiterer Vorteil der Software besteht in ihrer Integration in das Kassensystem des Geschäfts. Dadurch werden die Lagerbestände automatisch aktualisiert, sobald Verkäufe getätigt werden. Dies reduziert die Notwendigkeit manueller Bestandszählungen und minimiert das Risiko von Fehl- oder Überbeständen. Diese nahtlose Integration optimiert die Effizienz der Bestandsverwaltung und trägt zur Verbesserung der betrieblichen Abläufe bei.

Technisches Umfeld

Für den Webshop nutzen wir eine Vielzahl von Programmierumgebungen, darunter Visual Studio Code und gegebenenfalls Xcode. Unser Entwicklerteam arbeitet mit verschiedenen Programmiersprachen wie HTML, CSS, PHP, JavaScript, jQuery und NodeJS. Zur Datenhaltung setzen wir auf MySQL als Datenbanklösung. Zusätzlich verwenden wir verschiedene Frameworks wie Bootstrap, und phpunit, um eine solide Grundlage für die Entwicklung und das Design unserer Webanwendung zu schaffen.

Anforderungen

Scrum-Master (Chedjou):

Nicht funktionale Anforderung (Effektive Kommunikation):

- Der Scrum-Master muss sicherstellen, dass die Kommunikation im Team effektiv ist und alle Teammitglieder regelmäßig über den Projektstatus informiert sind.
- Bedingung für Anforderungsabnahme: Die Teammitglieder müssen mindestens zweimal pro Woche über den Projektstatus und eventuelle Hindernisse informiert werden, gemessen anhand von Team-Meetings und Statusberichten.
- Messmetrik und Durchführplan:

- Durchführung von wöchentlichen Team-Meetings, um den Projektfortschritt zu besprechen und Probleme zu lösen.
- Erstellung von Statusberichten und Verteilung an alle Teammitglieder.
- Überprüfung der Kommunikationseffektivität anhand von Umfragen und Feedback der Teammitglieder.

Funktionale Anforderung (Backlog-Management):

- Der Scrum-Master muss das Backlog effektiv verwalten und sicherstellen, dass alle Aufgaben priorisiert und zugewiesen werden.
- Bedingung für Anforderungsabnahme: Das Backlog muss wöchentlich aktualisiert und überprüft werden, und alle Aufgaben müssen den entsprechenden Teammitgliedern zugewiesen werden.
- Messmetrik und Durchführplan:
 - Überprüfung des Backlogs auf Vollständigkeit und Aktualität.
 - Zuteilung von Aufgaben anhand der Teamkapazität und Fähigkeiten.
 - Überprüfung der Aufgabenverteilung und Priorisierung bei wöchentlichen Backlog-Reviews.

Nicht funktionale Anforderung (Hindernisbeseitigung):

- Der Scrum-Master muss Hindernisse und Probleme im Projekt identifizieren und beseitigen, um den Projektfortschritt sicherzustellen.
- Bedingung für Anforderungsabnahme: Hindernisse müssen innerhalb von 24 Stunden nach ihrer Identifizierung beseitigt werden, gemessen anhand von Hindernis-Tracking und Lösungszeiten.
- Messmetrik und Durchführplan:
 - Erfassung von Hindernissen und Problemen in einem Hindernis-Tracking-System.
 - Priorisierung und Beseitigung von Hindernissen durch das Scrum-Master und das Entwicklungsteam.
 - Überprüfung der Lösungszeiten und Erfolgsraten bei der Hindernisbeseitigung.

Funktionale Anforderung (Sprint-Planung):

- Der Scrum-Master muss Sprint-Planungstreffen organisieren und sicherstellen, dass die Ziele des Sprints klar definiert und das Backlog entsprechend aktualisiert werden.

- Bedingung für Anforderungsabnahme: Die Sprint-Planungstreffen müssen vor Beginn jedes Sprints stattfinden und alle Teammitglieder müssen über die Ziele und Aufgaben informiert sein.
- Messmetrik und Durchführplan:
 - Durchführung von Sprint-Planungstreffen gemäß dem Scrum-Framework.
 - Klare Definition der Sprint-Ziele und Aufgaben.
 - Überprüfung der Teilnahme und Informiertheit der Teammitglieder bei Sprint-Planungstreffen.

Frontend-Entwickler (Knaus)

Nicht-funktionale Anforderung (Benutzerfreundlichkeit)

- Anforderung: Die Benutzeroberfläche der Webseite muss intuitiv und benutzerfreundlich gestaltet sein, um eine einfache Navigation und Interaktion für Benutzer jeden Alters zu ermöglichen.
 - Jugendliche (13-18 Jahre): Die Hauptzielgruppe sollte eine ansprechende und moderne Benutzeroberfläche vorfinden, die ihre Aufmerksamkeit gewinnt und eine intuitive Navigation ermöglicht.
 - Junge Erwachsene (18-25 Jahre): Die Benutzeroberfläche sollte flexibel genug sein, um deren Bedürfnisse zu erfüllen.
- Bedingung für Anforderungsabnahme: 90% der Testteilnehmer müssen die Webseite als benutzerfreundlich empfinden, gemessen anhand einer standardisierten Umfrage nach dem Test.
- Messmetrik und Durchführplan:
 - Durchführung von Benutzerinteraktionstests mit verschiedenen Aufgaben-Szenarien:
 - Registrierung: Testteilnehmer registrieren sich auf der Website. Beobachtung und Feedback zum Registrierungsprozess.
 - Produktsuche: Testteilnehmer suchen nach einem bestimmten Produkt und bewerten die Suchergebnisse.
 - Produktansicht: Testteilnehmer prüfen Produktdetails und bewerten die Präsentation.
 - Warenkorb und Checkout: Testteilnehmer legen Produkte in den Warenkorb und schließen den Checkout ab. Feedback zur Benutzererfahrung wird gesammelt.

- Kontaktieren des Kundensupports: Testteilnehmer kontaktieren den Support und bewerten die Zugänglichkeit und Reaktionszeit.
- Aktualisierung von Profilinformationen: Testteilnehmer aktualisieren Profilinformationen und geben Feedback zur Benutzerfreundlichkeit.
- Auswertung der Umfrageergebnisse und Anpassung der Benutzeroberfläche basierend auf dem Feedback.

Funktionale Anforderung (Produktfilterung)

- Anforderung: Die Produkte auf der Webseite müssen nach verschiedenen Kategorien gefiltert werden können.
 - Produktart: Hauptkategorien wie Kleidung, Elektronik, Möbel, etc.
 - Unterkategorien: Weitere Verfeinerung innerhalb der Hauptkategorien, z.B. Hemden, Hosen, Kleider bei Kleidung.
 - Marken oder Hersteller: Filterung nach bestimmten Marken oder Herstellern.
 - Preisspanne: Filterung nach einem bestimmten Preisbereich.
 - Größe: Filterung nach verschiedenen Größen, besonders relevant für Kleidung und Möbel.
- Bedingung für Anforderungsabnahme: Alle Filteroptionen müssen korrekt funktionieren und die entsprechenden Produkte gemäß den ausgewählten Kriterien anzeigen.
- Messmetrik und Durchführplan:
 - Testen der Filteroptionen mit einer Testdatenbank von Produkten unterschiedlicher Kategorien, Preise und Größen.
 - Durchführung von 15 Testläufen mit verschiedenen Filterkombinationen, z.B.:
 - Kategorie + Preisbereich
 - Kategorie + Marke
 - Kategorie + Größe
 - Preisbereich + Marke
 - Preisbereich + Größe
 - Marke + Größe
 - Kategorie + Preisbereich + Marke
 - Kategorie + Preisbereich + Größe
 - Kategorie + Marke + Größe
 - Preisbereich + Marke + Größe

- Überprüfung der korrekten Produktanzeige entsprechend den ausgewählten Filtern.

Nicht-funktionale Anforderung (Browserkompatibilität)

- Anforderung: Die Webseite muss in den gängigen Webbrowsern (Chrome, Firefox, Safari, Edge) korrekt angezeigt und voll funktionsfähig sein.
- Bedingung für Anforderungsabnahme: Die Webseite muss in mindestens 95% der Testsitzungen in jedem der genannten Browser korrekt geladen und ohne Anzeigefehler funktionieren.
- Messmetrik und Durchführplan:
 - Durchführung von 10 Testläufen in jedem Browser, auf verschiedenen Geräten und Betriebssystemen.
 - Dokumentation von Anzeigefehlern oder Inkompatibilitäten und deren Behebung durch das Entwicklungsteam.
 - Endgültige Überprüfung der Browserkompatibilität vor der Veröffentlichung der Webseite.

Funktionale Anforderung (Warenkorb-Funktionalität)

- Anforderung: Die Webseite muss eine Warenkorb-Funktionalität bereitstellen, die es Benutzern ermöglicht, Artikel hinzuzufügen, zu bearbeiten und zu entfernen.
- Bedingung für Anforderungsabnahme: Der Warenkorb muss korrekt aktualisiert werden, und alle Änderungen müssen in Echtzeit durch Ajax-Anfragen auf der Webseite angezeigt werden.
- Messmetrik und Durchführplan:
 - Durchführung von 50 Testläufen mit verschiedenen Szenarien, wie dem Hinzufügen, Bearbeiten und Entfernen von Artikeln im Warenkorb.
 - Überprüfung der Aktualisierung des Warenkorbs in Echtzeit und der korrekten Anzeige der Gesamtsumme, z.B.:
 - Hinzufügen von Produkten: Produkt hinzufügen und überprüfen, ob die Gesamtsumme in Echtzeit aktualisiert wird.
 - Entfernen von Produkten: Produkt entfernen und überprüfen, ob die Gesamtsumme entsprechend reduziert wird.
 - Aktualisierung von Produktmengen: Menge ändern und überprüfen, ob die Gesamtsumme sofort aktualisiert wird.

- Simulation von Transaktionen und Überprüfung der Zuverlässigkeit der Warenkorb-Funktionalität.

Backend-Entwickler (Steshkov):

Nicht funktionale Anforderung (Datensicherheit):

- Anforderung: Die Backend-Infrastruktur muss robuste Sicherheitsmaßnahmen implementieren, um die Vertraulichkeit und Integrität der Benutzerdaten zu gewährleisten. Dazu gehören:
 1. Datenverschlüsselung während der Übertragung und Speicherung.
 2. Granulare Zugriffskontrolle und Berechtigungsverwaltung.
 3. Regelmäßige Sicherheitsupdates und Patch-Management.
 4. Logging und Überwachung verdächtiger Aktivitäten.
 5. Regelmäßige Daten-Backups und Notfallwiederherstellung.
 6. Sicherheitsbewusstseinschulungen für Mitarbeiter.
 7. Durchführung von Penetrationstests und Sicherheitsaudits.
- Bedingung für Anforderungsabnahme: Die Sicherheitsmaßnahmen müssen den geltenden Datenschutzrichtlinien entsprechen und von einem unabhängigen Sicherheitsexperten überprüft werden.
- Messmetrik und Durchführplan:
 - Durchführung einer Sicherheitsüberprüfung durch einen externen Sicherheitsexperten (Tester), um potenzielle Sicherheitslücken und Schwachstellen zu identifizieren.
 - Implementierung von Sicherheitspatches und Aktualisierungen basierend auf den Empfehlungen des Sicherheitsexperten.
 - Endgültige Überprüfung der Backend-Sicherheit vor der Bereitstellung der Webseite.

Funktionale Anforderung (Datenbankoptimierung):

- Anforderung: Die Datenbankverbindung sollte so optimiert sein, dass Produktinformationen und Bestandsdaten schnell und effizient abgerufen werden können, idealerweise innerhalb von wenigen Millisekunden.
- Bedingung für Anforderungsabnahme: Die Datenbankabfragen müssen innerhalb von maximal 2 Sekunden abgeschlossen sein, gemessen anhand einer standardisierten Benchmark-Testsuite.
- Messmetrik und Durchführplan:

- Durchführung von Leistungstests mit einer repräsentativen MySQL-Datenbank von Produkten und Bestandsdaten.
- Überprüfung der Antwortzeiten und Optimierung der Datenbankabfragen bei Bedarf.
- Endgültige Überprüfung der Datenbankleistung vor der Veröffentlichung der Webseite.

Nicht funktionale Anforderung (Skalierbarkeit):

- Anforderung: Die Backend-Infrastruktur muss skalierbar sein, um mit einem wachsenden Benutzer- und Transaktionsvolumen umzugehen, ohne dass die Leistung oder Verfügbarkeit des Systems beeinträchtigt wird.
- Bedingung für Anforderungsabnahme: Die Backend-Infrastruktur muss unter Last getestet werden und in der Lage sein, zusätzliche Ressourcen bei Bedarf automatisch zu skalieren.
- Messmetrik und Durchführplan:
 - Durchführung von Lasttests mit simulierten Benutzerzugriffen und Transaktionen.
 - Überprüfung der Reaktion der Infrastruktur auf steigende Lasten und automatische Skalierung bei Bedarf.
 - Endgültige Überprüfung der Skalierbarkeit vor der Veröffentlichung der Webseite.

Tester (Zamanova):

Nicht funktionale Anforderung (Softwarequalität):

- Anforderung: Der Tester muss die Softwarequalität durch umfassende Tests sicherstellen und sicherstellen, dass alle Anforderungen erfüllt sind.
- Bedingung für Anforderungsabnahme: Die Software muss mindestens 95% der Testfälle bestehen und alle funktionalen und nicht funktionalen Anforderungen erfüllen.
- Messmetrik und Durchführplan:
 - Erstellung von Testplänen und Testfällen basierend auf den Anforderungen.
 - Durchführung von manuellen und automatisierten Tests in verschiedenen Umgebungen und Szenarien.
 - Dokumentation von Testergebnissen und Fehlerberichten.

Nicht funktionale Anforderung (Benutzererfahrung):

- Anforderung: Der Tester muss die Benutzererfahrung der Webseite überprüfen und sicherstellen, dass sie benutzerfreundlich und intuitiv ist.
- Bedingung für Anforderungsabnahme: Die Webseite muss von mindestens 80% der Testteilnehmer als benutzerfreundlich bewertet werden, gemessen anhand von Umfragen und Feedback.
- Messmetrik und Durchführplan:
 - Durchführung von Benutzer-Tests mit repräsentativen Testpersonen unterschiedlichen Alters und Hintergrunds (15 bis 20 Personen).
 - Sammeln von Feedback und Bewertungen zur Benutzerfreundlichkeit der Webseite.
 - Anpassung der Benutzeroberfläche basierend auf dem gesammelten Feedback.

Nicht funktionale Anforderung (Datenbankabfragen):

- Anforderung: Der Tester muss die Effizienz der Datenbankabfragen überprüfen und sicherstellen, dass sie schnell und zuverlässig sind.
- Bedingung für Anforderungsabnahme: Die Datenbankabfragen müssen innerhalb von maximal 2 Sekunden abgeschlossen sein und korrekte Ergebnisse liefern.
- Messmetrik und Durchführplan:
 - Durchführung von Leistungstests mit einer repräsentativen Datenbank von Produkten und Bestandsdaten.
 - Überprüfung der Antwortzeiten und Optimierung der Datenbankabfragen bei Bedarf.
 - Identifizierung und Dokumentation von Leistungsproblemen und deren Behebung durch das Entwicklungsteam.

Nicht funktionale Anforderung (Skalierbarkeit):

- Anforderung: Der Tester muss die Skalierbarkeit der Webseite überprüfen und sicherstellen, dass sie auch bei steigender Nutzerzahl effizient funktioniert.
- Bedingung für Anforderungsabnahme: Die Webseite muss unter Last getestet werden und in der Lage sein, zusätzliche Benutzer ohne Leistungseinbußen zu unterstützen.
- Messmetrik und Durchführplan:
 - Durchführung von Lasttests mit simulierten Benutzerzugriffen und Transaktionen.
 - Überprüfung der Reaktion der Webseite auf steigende Lasten und automatische Skalierung bei Bedarf.

- Identifizierung und Dokumentation von Leistungsproblemen und deren Behebung durch das Entwicklungsteam.

Nicht funktionale Anforderung (Automatisierte Tests für die Funktionalität des Warenkorbs):

- Anforderung: Der Tester muss automatisierte Tests für die Funktionalität des Warenkorbs implementieren.
- Bedingung für Anforderungsabnahme: Alle Testfälle für das Hinzufügen, Bearbeiten und Entfernen von Artikeln im Warenkorb müssen erfolgreich durchgeführt werden.
- Messmetrik und Durchführplan:
 - Testfall: Hinzufügen von Artikeln
 - Aktion: Füge einen Artikel zum Warenkorb hinzu.
 - Erwartetes Ergebnis: Der Artikel wird erfolgreich zum Warenkorb hinzugefügt.
 - Testfall: Bearbeiten von Artikeln
 - Aktion: Bearbeite die Menge oder Eigenschaften eines Artikels im Warenkorb.
 - Erwartetes Ergebnis: Die Änderungen werden korrekt im Warenkorb angezeigt.
 - Testfall: Entfernen von Artikeln
 - Aktion: Entferne einen Artikel aus dem Warenkorb.
 - Erwartetes Ergebnis: Der Artikel wird erfolgreich aus dem Warenkorb entfernt.
 - Durchführung von Testszenarien:
 - Führe jeden Testfall für das Hinzufügen, Bearbeiten und Entfernen von Artikeln im Warenkorb durch.
 - Überprüfe nach jeder Aktion die korrekte Aktualisierung des Warenkorbs und der Gesamtsumme.
 - Identifiziere und dokumentiere Fehler oder Unregelmäßigkeiten, falls welche auftreten.

Funktionale Anforderung (Erstellung eines Testautomatisierungs-Tools):

- Anforderung: Der Tester muss ein Testautomatisierungs-Tool entwickeln, das die Durchführung und Auswertung von Testszenarien erleichtert.

- Bedingung für Anforderungsabnahme: Das Tool muss in der Lage sein, mindestens 90% der definierten Testszenarien automatisiert durchzuführen und die Ergebnisse korrekt zu protokollieren.
- Messmetrik und Durchführplan:
 - Entwicklung des Testautomatisierungs-Tools basierend auf den definierten Testszenarien.
 - Implementierung von Funktionen zur automatisierten Durchführung und Auswertung von Tests.
 - Durchführung von Pilottests zur Validierung der Funktionalität des Tools.
 - Dokumentation und Behebung von Fehlern oder Optimierungspotenzialen im Tool.

JIRA Link:

<https://voguevibes.atlassian.net/jira/software/projects/SCRUM/boards/1?atlOrigin=eyJpIjoiNzU1ODE1Zjk4ZmMzMzNGM3YzkyMjA3ZDBkZjg5YjIwMmliLCJwIjoiajI9>

Wir haben uns für den Custom Clothing Store entschieden, weil wir der Überzeugung sind, dass Individualität in der Mode von großer Bedeutung ist. Hier können wir unsere persönlichen Vorstellungen und Stilpräferenzen direkt in die Gestaltung unserer Kleidung einfließen lassen. Das gibt uns die Möglichkeit, nicht nur einzigartige Stücke zu tragen, sondern auch solche, die perfekt auf unsere Maße und Bedürfnisse zugeschnitten sind. Diese persönliche Note in unserer Garderobe zu haben, macht unsere Outfits nicht nur besonders, sondern verstärkt auch unser Selbstbewusstsein im Alltag. Mit Jira können wir Aufgaben effizient verwalten und den Projektfortschritt in Echtzeit überwachen, was für die Koordinierung unserer kreativen und logistischen Prozesse unerlässlich ist.

GITHUB Link:

<https://github.com/orgs/VogueVibes/dashboard>

ZEITPLAN:

Spin 1: Vorbereitung und Planung

- Dauer: 10.05.2024
- Aufgaben:
 - Scrum-Master:
 - Durchführung des Sprint 0, Einrichtung des Projekts und Erstellung des Backlogs.
 - Schulungen für das Team zur Vertrautheit mit dem Scrum-Prozess.

- Frontend-Entwickler:
 - Einrichtung der Entwicklungsumgebung und Konfiguration von Tools wie Jira.
 - Vorbereitung der Frontend-Entwicklung.
- Backend-Entwickler:
 - Evaluierung der Backend-Infrastruktur und Planung der Implementierung.
 - Vorbereitung der Backend-Entwicklung.
- Tester:
 - Erstellung von Testplänen und Vorbereitung auf kommende Testszenarien.

Spin 2: Entwicklung und Design

- Dauer: 2.06.2024
- Aufgaben:
 - Scrum-Master:
 - Überwachung des Projektfortschritts und Lösung von Hindernissen.
 - Durchführung von wöchentlichen Team-Meetings.
 - Frontend-Entwickler:
 - Implementierung der Benutzeroberfläche gemäß den Designspezifikationen.
 - Entwicklung von Funktionalitäten wie Produktfilterung.
 - Backend-Entwickler:
 - Implementierung der Backend-Infrastruktur, einschließlich Sicherheitsmaßnahmen und API-Integrationen.
 - Tester:
 - Durchführung von Tests für die entwickelten Frontend-Funktionen.

Spin 3: Integration und Optimierung

- Dauer: 15.06.24
- Aufgaben:
 - Scrum-Master:
 - Sicherstellung der Effektivität der Kommunikation und Lösung von Hindernissen.
 - Frontend-Entwickler:
 - Integration der Frontend-Komponenten und Anpassungen basierend auf Tester-Feedback.
 - Backend-Entwickler:

- Optimierung der Datenbankabfragen und Sicherheitsmaßnahmen.
- Tester:
 - Durchführung von Tests für integrierte Frontend- und Backend-Komponenten.

Spin 4: Testing und Fehlerbehebung

- Dauer: 20.06.24
- Aufgaben:
 - Scrum-Master:
 - Organisation von Sprint-Planungstreffen und Überprüfung der Sprint-Ziele.
 - Frontend-Entwickler:
 - Behebung von Frontend-Fehlern und Anpassungen basierend auf Tester-Feedback.
 - Backend-Entwickler:
 - Behebung von Backend-Fehlern und Sicherstellung der Datenbankoptimierung.
 - Tester:
 - Durchführung umfassender Tests, um Fehler zu identifizieren und zu dokumentieren.

Spin 5: Abschluss und Bereitstellung

- Dauer: 27.06.2024
- Aufgaben:
 - Scrum-Master:
 - Abschlussprüfung aller Funktionen und Nichtfunktionalitäten.
 - Vorbereitung der Veröffentlichung und Bereitstellung der Webseite.
 - Frontend-Entwickler und Backend-Entwickler:
 - Letzte Anpassungen und Fehlerbehebungen basierend auf dem abschließenden Tester-Feedback.
 - Tester:
 - Überprüfung der abschließenden Version der Webseite und Genehmigung für die Veröffentlichung.

Kommunikation und Meetings (Scrum-Master)

Ich, als Scrum-Master habe eine WhatsApp-Gruppe erstellt, in der wir unsere Fortschritte dokumentieren und uns bei Fragen gegenseitig unterstützen. Jeden Sonntag um 17 Uhr gibt es ein kurzes Meeting, um den neuesten Stand zu besprechen. Zudem findet zu jedem Sprint mindestens ein 30-minütiges Meeting über Microsoft Teams statt, um die Sprint-Ziele und Aufgaben zu definieren und sicherzustellen, dass alle Teammitglieder informiert sind.

Als Scrum-Master Sorge ich für eine effektive Kommunikation im Team. Dies umfasst folgende Maßnahmen:

1. Wöchentliche Team-Meetings:

- Durchführung von wöchentlichen Team-Meetings zur Besprechung des Projektfortschritts und zur Lösung von Problemen.
- Erstellung von Statusberichten und Verteilung an alle Teammitglieder, um sicherzustellen, dass alle über den aktuellen Stand informiert sind.

2. Verwendung von JIRA:

- Verwaltung und Nachverfolgung des Backlogs in JIRA, um eine transparente und organisierte Aufgabenverteilung zu gewährleisten.
- Das Backlog wird wöchentlich aktualisiert und überprüft, wobei alle Aufgaben den entsprechenden Teammitgliedern zugewiesen werden.
- Regelmäßige Überprüfung (wöchentlich) des Backlogs auf Vollständigkeit und Aktualität.

<input type="checkbox"/> Sprinttafel 1 20 May – 31 May (4 issues)	0 0 0	Complete sprint	...
Während des ersten Sprints müssen wir die in der Sprintliste aufgeführten Aufgaben zusammenfassen. Wir werden auch über die bisherigen Fehler und Unzulänglichkeiten sprechen und über den zweit...			
<input checked="" type="checkbox"/> SCRUM-3	Seitenlayouts erstellen (Figma)	DONE ▾	- [Avatar]
<input checked="" type="checkbox"/> SCRUM-4	Registrierung und Login-Funktionalität	DONE ▾	- AK
<input checked="" type="checkbox"/> SCRUM-14	Dokumentieren und Nachfragen	DONE ▾	- CA
<input checked="" type="checkbox"/> SCRUM-23	Planung für den nächsten Sprint	DONE ▾	- CA

<input type="checkbox"/> Sprinttafel 2 31 May – 2 Jun (7 issues)	0	0	0	Complete sprint	...
<input checked="" type="checkbox"/> SCRUM-5 Homepage			DONE	-	CA
<input checked="" type="checkbox"/> SCRUM-16 Erstellen einer Navigationsleiste für alle Seiten einer Website			DONE	-	
<input checked="" type="checkbox"/> SCRUM-17 Erstellung von Katalogseiten			DONE	-	
<input checked="" type="checkbox"/> SCRUM-18 Erstellen einer „About Us“-Seite			DONE	-	AK
<input checked="" type="checkbox"/> SCRUM-19 Erstellen einer Hilfe-Seite			DONE	-	AK
<input checked="" type="checkbox"/> SCRUM-20 Prüfung und Fehlerkorrektur			DONE	-	NZ
<input checked="" type="checkbox"/> SCRUM-22 Dokumentieren und Nachfragen			DONE	-	CA

<input type="checkbox"/> Sprinttafel 3 2 Jun – 15 Jun (5 issues)	0	0	0	Complete sprint	...
<input checked="" type="checkbox"/> SCRUM-27 Profilentwicklung			DONE	-	
<input checked="" type="checkbox"/> SCRUM-24 Unit Test			DONE	-	NZ
<input checked="" type="checkbox"/> SCRUM-40 Anpassung der Website für verschiedene Bildschirme			DONE	-	AK
<input checked="" type="checkbox"/> SCRUM-42 Hinzufügen zusätzlicher Funktionen			DONE	-	
<input checked="" type="checkbox"/> SCRUM-41 Dokumentenerstellung			DONE	-	CA

Projects / VogueVibes

Backlog

Q Search

CA

AK

NZ

Элик ▾

Insights

View settings

>

Sprinttafel 1

20 May – 31 May

(4 issues)

0

0

0

Complete sprint

...

>

Sprinttafel 2

31 May – 2 Jun

(7 issues)

0

0

0

Complete sprint

...

>

Sprinttafel 3

2 Jun – 15 Jun

(5 issues)

0

0

0

Complete sprint

...

>

Sprinttafel 4

20 Jun – 27 Jun

(6 issues)

0

0

0

Complete sprint

...

Projects / VogueVibes

List

Search list

AK

CA

NZ

Share

Filter

Group

More

<input type="checkbox"/>	Type	# Key	Summary	Status	Sprint	Assignee	
<input type="checkbox"/>	> <input checked="" type="checkbox"/>	SCRUM-27	Profilentwicklung	DONE	Sprinttafel 3	<div><div></div><div>Vadim</div></div>	
<input type="checkbox"/>	> <input checked="" type="checkbox"/>	SCRUM-24	Unit Test	DONE	Sprinttafel 3	<div><div></div><div>Nina Z</div></div>	
<input type="checkbox"/>	> <input checked="" type="checkbox"/>	SCRUM-3	Seitenlayouts erstellen (Figma)	DONE	Sprinttafel 1	<div><div></div><div>Vadim</div></div>	
<input type="checkbox"/>	> <input checked="" type="checkbox"/>	SCRUM-4	Registrierung und Login-Funktionalität	DONE	Sprinttafel 1	<div><div></div><div>Anastasiia Knaus</div></div>	
<input type="checkbox"/>	> <input checked="" type="checkbox"/>	SCRUM-5	Homepage	DONE	Sprinttafel 2	<div><div></div><div>Chedjou Ange</div></div>	
<input type="checkbox"/>	> <input checked="" type="checkbox"/>	SCRUM-14	Dokumentieren und Nachfragen	DONE	Sprinttafel 1	<div><div></div><div>Chedjou Ange</div></div>	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SCRUM-23	Planung für den nächsten Sprint	DONE	Sprinttafel 1	<div><div></div><div>Chedjou Ange</div></div>	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SCRUM-40	Anpassung der Website für verschiedene Bildschirme	DONE	Sprinttafel 3	<div><div></div><div>Anastasiia Knaus</div></div>	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SCRUM-16	Erstellen einer Navigationsleiste für alle Seiten einer Webs...	DONE	Sprinttafel 2	<div><div></div><div>Vadim</div></div>	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SCRUM-17	Erstellung von Katalogseiten	DONE	Sprinttafel 2	<div><div></div><div>Vadim</div></div>	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SCRUM-18	Erstellen einer „About Us“-Seite	DONE	Sprinttafel 2	<div><div></div><div>Anastasiia Knaus</div></div>	

List

Give feedback

<input type="checkbox"/>	Type	# Key	Summary	Status	Sprint	Assignee	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SCRUM-18	Erstellen einer „About Us“-Seite	DONE	Sprinttafel 2	AK Anastasiia Knaus	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SCRUM-51	Hochladen von Code auf Github	IN PROGRESS	Sprinttafel 4	CA Chedjou Ange	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SCRUM-42	Hinzufügen zusätzlicher Funktionen	DONE	Sprinttafel 3	Vadim	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SCRUM-41	Dokumentenerstellung	DONE	Sprinttafel 3	CA Chedjou Ange	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SCRUM-47	FunktionsTest	DONE	Sprinttafel 4	Vadim	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SCRUM-19	Erstellen einer Hilfe-Seite	DONE	Sprinttafel 2	AK Anastasiia Knaus	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SCRUM-20	Prüfung und Fehlerkorrektur	DONE	Sprinttafel 2	NZ Nina Z	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SCRUM-49	Fehlerkorrektur	DONE	Sprinttafel 4	NZ Nina Z	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	SCRUM-22	Dokumentieren und Nachfragen	DONE	Sprinttafel 2	CA Chedjou Ange	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SCRUM-48	Befragung von 15 Personen	DONE	Sprinttafel 4	CA Chedjou Ange	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SCRUM-50	Verbindung der Software-Tile zu einem Ganzen	DONE	Sprinttafel 4	Vadim	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SCRUM-52	Vorbereitung der endgültigen Dokumentationen und Anfor...	IN PROGRESS	Sprinttafel 4	CA Chedjou Ange	

<input type="checkbox"/>	Sprinttafel 4 20 Jun – 27 Jun (6 issues)	0 0 0	Complete sprint	...
<input checked="" type="checkbox"/>	SCRUM-51 Hochladen von Code auf Github	IN PROGRESS	CA	
<input checked="" type="checkbox"/>	SCRUM-47 FunktionsTest	DONE	Vadim	
<input checked="" type="checkbox"/>	SCRUM-49 Fehlerkorrektur	DONE	NZ	
<input checked="" type="checkbox"/>	SCRUM-48 Befragung von 15 Personen	DONE	CA	
<input checked="" type="checkbox"/>	SCRUM-50 Verbindung der Software-Tile zu einem Ganzen	DONE	Vadim	
<input checked="" type="checkbox"/>	SCRUM-52 Vorbereitung der endgültigen Dokumentationen und Anforderungen	IN PROGRESS	CA	

3. Hindernisbeseitigung:

- Identifizierung und Beseitigung von Hindernissen und Problemen im Projekt innerhalb von 24 Stunden nach ihrer Identifizierung.
- Priorisierung und Beseitigung von Hindernissen in Zusammenarbeit mit dem Entwicklungsteam.

4. Sprint-Planung:

- Organisation von Sprint-Planungstreffen vor Beginn jedes Sprints.
- Klare Definition der Sprint-Ziele und Aufgaben.
- Sicherstellung, dass alle Teammitglieder über die Ziele und Aufgaben des Sprints informiert sind.
- Überprüfung der Teilnahme und Informiertheit der Teammitglieder bei Sprint-Planungstreffen.

5. Effektive Kommunikation:

- Sicherstellung, dass die Kommunikation im Team effektiv ist und alle Teammitglieder regelmäßig über den Projektstatus und eventuelle Hindernisse informiert sind.
- Durchführung von Umfragen und Einholung von Feedback der Teammitglieder zur Überprüfung der Kommunikationseffektivität.

Diese Maßnahmen haben mir dabei geholfen, dass alle Teammitglieder stets über den Projektfortschritt informiert sind und effektiv zusammenarbeiten, um die Projektziele zu erreichen.

Frontend

Die Implementierung der Frontendkomponenten einer Webanwendung wurde erfolgreich abgeschlossen. Dazu gehört die Login-Seite, die Registrierungsseite und die Startseite. Jede Seite verwendet HTML, CSS und JavaScript, um eine interaktive und benutzerfreundliche Schnittstelle zu schaffen. AJAX/Fetch API wird verwendet, um Daten asynchron an den Server zu senden und Ergebnisse zu erhalten.

Login-Seite

Benutzerinteraktion:

1. Benutzer gibt E-Mail und Passwort ein.
2. Benutzer klickt auf den "Login"-Button.
3. Anmeldedaten werden an den Server gesendet.
4. Das Ergebnis wird angezeigt (erfolgreiche Anmeldung oder Fehler).

Client-seitige Validierung:

1. Überprüfung des E-Mail-Formats mittels regulären Ausdrucks.
2. Überprüfung, dass das Passwortfeld nicht leer ist.

Verwendete Technologien:

1. HTML: Struktur der Login-Seite.
2. CSS: Styling der Login-Seite.
3. JavaScript: Logik für die Interaktionen und Validierungen.
4. AJAX/Fetch API: Senden der Anmeldedaten an den Server und Verarbeiten der Antwort.

Registrierungsseite

Benutzerinteraktion:

1. Benutzer gibt Registrierungsdaten ein (E-Mail, Passwort, Passwort bestätigen).

2. Benutzer klickt auf den "Sign Up"-Button.
3. Registrierungsdaten werden an den Server gesendet.
4. Das Ergebnis wird angezeigt (erfolgreiche Registrierung oder Fehler).

Client-seitige Validierung:

1. Überprüfung des E-Mail-Formats.
2. Überprüfung, dass die Passwörter übereinstimmen.

Verwendete Technologien:

1. HTML: Struktur der Registrierungsseite.
2. CSS: Styling der Registrierungsseite.
3. JavaScript: Logik für die Interaktionen und Validierungen.
4. AJAX/Fetch API: Senden der Registrierungsdaten an den Server und Verarbeiten der Antwort

Startseite

Benutzerinteraktion:

1. Anzeige einer Willkommensnachricht und Produktbannern.
2. Benutzer wählt ein Produkt aus.
3. Produkt wird dem Warenkorb hinzugefügt.

Dynamische Inhaltsaktualisierung:

- Aktualisierung der Seite ohne Neu laden, z.B. Anzeige einer Fehlermeldung bei falscher Anmeldung oder Aktualisierung des Warenkorbs.

Verwendete Technologien:

1. HTML: Struktur der Startseite.
2. CSS: Styling der Startseite.
3. JavaScript: Logik für die Interaktionen und dynamische Inhaltsaktualisierungen.
4. AJAX/Fetch API: Kommunikation mit dem Server für Produktdaten und Warenkorbaktualisierungen

Dokumentation: Erfüllung der Anforderungen

Ergebnisse:

Nicht-funktionale Anforderung: Benutzerfreundlichkeit

Die Benutzerfreundlichkeitsanforderung wurde erfüllt und von den Testteilnehmern bestätigt.

Funktionale Anforderung: Produktfilterung

Die Filteroptionen funktionieren korrekt und zeigen die entsprechenden Produkte gemäß den ausgewählten Kriterien an.

Nicht-funktionale Anforderung: Browserkompatibilität

Die Webseite funktionierte in 98% der Testsitzungen in jedem der genannten Browser korrekt und ohne Anzeigefehler.

Funktionale Anforderung: Warenkorb-Funktionalität

Der Warenkorb wurde korrekt aktualisiert und alle Änderungen in Echtzeit angezeigt. Die Gesamtsumme wurde korrekt berechnet und angezeigt.

Alle aufgeführten Anforderungen wurden erfolgreich umgesetzt und die entsprechenden Tests haben bestätigt, dass die Anforderungen erfüllt wurden.

Backend

Nicht funktionale Anforderung (Datensicherheit):

Was wurde gemacht:

- Benutzerumfrage: Wir haben eine Umfrage unter unseren Bekannten durchgeführt, bei der 15 Personen teilgenommen haben, die aktiv verschiedene Webdienste nutzen. Diese Benutzer sind auch Studierende verschiedener Informatik-Fakultäten. Sie haben ihre Meinungen und ihr Feedback bezüglich der Sicherheit und Benutzerfreundlichkeit unserer Dienste mitgeteilt. Die erhaltenen Rückmeldungen haben uns geholfen, die Hauptbedenken der Benutzer und die wichtigsten Bereiche, die verbessert werden müssen, zu verstehen.
- Sicherheitsprüfung: Nach Erhalt des Feedbacks haben wir einen unabhängigen Sicherheitsexperten von der FH Technikum Wien, einen Studenten des 5. Semesters der Fakultät Informationstechnologie, eingeladen, um eine umfassende Sicherheitsprüfung unseres Backendsystems durchzuführen. Der Experte hat alle Sicherheitsaspekte unseres Systems manuell und gründlich überprüft, mehrere Schwachstellen aufgedeckt und entsprechende Maßnahmen zu deren Behebung vorgeschlagen. Nach der Überprüfung wurde ein positiver und ermutigender Bericht über den Sicherheitsstatus unseres Systems erstellt.
- Umsetzung der Sicherheitsmaßnahmen: Basierend auf den Empfehlungen des Experten haben wir verschiedene Maßnahmen zum Schutz der Daten implementiert. Diese Maßnahmen umfassten:
 - Verschlüsselung: Alle sensiblen Daten werden nun sowohl im Ruhezustand als auch während der Übertragung verschlüsselt.

- Schutz vor SQL-Injektionen: Wir haben den Code aktualisiert, um SQL-Injektionen zu verhindern, indem wir parametrisierte Abfragen und die Verwendung von ORM eingeführt haben.
- Schutz vor mehrfachen Sitzungen im selben Browser: Wir haben Mechanismen implementiert, um gleichzeitige Anmeldungen im System aus mehreren Tabs desselben Browsers zu verhindern, was hilft, Überlastungen zu vermeiden und vor DDoS-Angriffen zu schützen.
- Abschließende Prüfung: Vor der Inbetriebnahme des Systems haben wir eine abschließende Sicherheitsprüfung durchgeführt. Diese Prüfung bestätigte, dass alle entdeckten Schwachstellen behoben wurden und die Daten der Benutzer gemäß den geltenden Vorschriften geschützt sind.

Diese Maßnahmen haben es uns ermöglicht, das Sicherheitsniveau unseres Systems erheblich zu verbessern, den zuverlässigen Schutz der Benutzerdaten zu gewährleisten und allen notwendigen Standards und Anforderungen zu entsprechen.

Funktionale Anforderungen (Beispiel: Datenbankoptimierung)

Anforderung: Die Verbindung zur Datenbank muss optimiert werden, um schnelle und effiziente Abfragen von Produktinformationen und Bestandsdaten zu ermöglichen.

Abnahmekriterium: Datenbankabfragen müssen innerhalb von maximal 2 Sekunden abgeschlossen sein, gemessen anhand einer standardisierten Benchmark-Testsuite.

Metrik und Durchführungsplan:

- Durchführung von Leistungstests mit einer repräsentativen Datenbank von Produkten und Bestandsdaten.
- Überprüfung der Antwortzeiten und Optimierung der Datenbankabfragen bei Bedarf.
- Endgültige Überprüfung der Datenbankleistung vor der Veröffentlichung der Webseite.

Zur Erfüllung dieser Anforderungen:

- Automatisierte Tests: Es wurden automatisierte Tests in der Anzahl von 32 Stück geschrieben. Davon:

- 24 funktionale Tests: Diese Tests decken den gesamten Zyklus ab, beginnend mit der Registrierung und dem Login, bis hin zum Hinzufügen eines Produkts in den Warenkorb und dem Abschluss des Kaufs.
- 3 Modultests: Diese Tests konzentrieren sich auf das Login, auch bekannt als Benutzervalidierung.
- 5 Modultests zur Verifikation: Wir haben uns entschieden, Modultests für die Registrierung zu verwenden.
- Durchführung der Tests: Alle Tests wurden erfolgreich bestanden. Die Ausführungszeiten der Abfragen entsprachen den Anforderungen. Zum Beispiel betrug die Ausführungszeit einer Abfrage zur Registrierung und Anmeldung im Durchschnitt 0,8 Sekunden, die Zeit zum Hinzufügen eines Produkts in den Warenkorb 1,2 Sekunden und die Zeit für die Ausführung der Abfrage beim Checkout 1,9 Sekunden.
- Analyse und Optimierung: Die durchgeführte Analyse zeigte, dass alle Skripte, Algorithmen und Funktionen des Backends korrekt und effizient ausgeführt werden. Notwendige Optimierungen wurden vorgenommen, um eine stabile Leistung sicherzustellen. Wir haben Indizes in der Datenbank verwendet und komplexe Abfragen optimiert, was die Ausführungszeit erheblich verkürzt hat.
- Berücksichtigung der Abfragekomplexität: Es ist wichtig zu beachten, dass die Ausführungszeiten der Abfragen je nach Komplexität variieren. Zum Beispiel beinhalten Checkout-Abfragen viele Operationen wie die Überprüfung der Warenverfügbarkeit, die Berechnung der Gesamtkosten und die Zahlungsabwicklung, daher benötigen sie mehr Zeit. Andererseits werden Login-Abfragen deutlich schneller ausgeführt, da sie weniger ressourcenintensiv sind.
- Datenbankoptimierung: Statt Caching wurden Indizierung und Abfrageoptimierung angewendet. Indizes wurden zu häufig verwendeten Feldern wie username, email in der Tabelle user und user_id in der Tabelle basket hinzugefügt. Dies verbesserte die Antwortzeiten der Abfragen erheblich.

Somit kann ich als Backend-Entwickler mit Sicherheit sagen, dass alle Funktionen einwandfrei funktionieren und die Abfragen sehr schnell ausgeführt werden. Die Unterschiede in den Ausführungszeiten der Abfragen hängen von ihrer Komplexität ab, aber alle werden innerhalb des festgelegten Limits von 2 Sekunden ausgeführt, was den Anforderungen vollständig entspricht.

Zur Erfüllung dieser Anforderungen:

- **Integrationstests:** Es wurden Integrationstests mit externen Diensten entwickelt und durchgeführt. Diese Tests überprüften die korrekte Verarbeitung und Übertragung von Daten zwischen den Systemen. Wir testeten die Interaktion mit mehreren wichtigen Diensten, Systemen zur Generierung von Tracking-Nummern für Waren und externen Datenbanken.
- **Simulation von Transaktionen:** Es wurden Simulationen verschiedener Transaktionstypen durchgeführt, um die korrekte Datenverarbeitung zu gewährleisten. Die Tests umfassten die Erstellung von Bestellungen, die Zahlungsabwicklung und die Aktualisierung des Bestellstatus. Alle Transaktionen wurden erfolgreich abgeschlossen, die Daten wurden korrekt übertragen und verarbeitet. Benutzer können in ihren Profilen gekaufte Waren sehen und diese mithilfe der Tracking-Nummern verfolgen. Die Tracking-Codes werden dank eines speziellen Skripts immer einzigartig generiert.
- **Fehlererkennung und -behebung:** Während der Tests wurden mehrere Fehler entdeckt und behoben, wie zum Beispiel die fehlerhafte Ausnahmebehandlung und Probleme mit dem Datenformat. Dies erhöhte die Zuverlässigkeit und Stabilität des Systems.
- **Leistungsoptimierung:** Es wurden Leistungstests durchgeführt, um die Geschwindigkeit der Anfrageverarbeitung zu bewerten. Wir haben die Anfragen optimiert, Verzögerungen minimiert und eine schnelle und effiziente Datenverarbeitung sichergestellt. Die durchschnittliche Antwortzeit wurde auf 0,5 Sekunden für Standardoperationen und auf 1,2 Sekunden für komplexere Transaktionen wie die Zahlungsabwicklung reduziert.
- **Endgültige Überprüfung:** Vor dem Start des Systems wurde eine endgültige Überprüfung aller Integrationen durchgeführt. Diese Überprüfung bestätigte, dass alle Integrationen korrekt funktionieren, die Daten fehlerfrei übertragen und verarbeitet werden und die Leistung den Erwartungen entspricht.

Obwohl der Schwerpunkt auf der Integration für den Checkout-Prozess lag, stellt der Erfolg dieser Integrationen eine zuverlässige Grundlage für die weitere Erweiterung der Systemfunktionen dar. Wir planen, an der Optimierung weiterer Schlüsselfunktionen wie dem Benutzerprofil-Management und der Rücksendungsabwicklung weiterzuarbeiten, um eine noch höhere Leistung und Stabilität des Systems zu gewährleisten.

Tester

Die durchgeführten Tests der Software haben gezeigt, dass die ursprünglichen Anforderungen größtenteils erfüllt wurden. Sowohl funktionale als auch nicht funktionale Anforderungen wurden umfassend geprüft, um die Softwarequalität sicherzustellen.

Anforderung	Beschreibung	Erfüllungskriterium	Testergebnisse	Status
Software-qualität	Sicherstellen der Softwarequalität	Mindestens 95% der Testfälle bestanden	99% der Testfälle bestanden, alle Anforderungen erfüllt	Erfüllt
Funktionalität des Warenkorbs	Überprüfen der korrekten Funktionalität des Warenkorbs	Alle Aktionen korrekt ausgeführt, keine Fehler	Alle Aktionen werden korrekt ausgeführt, keine Fehler oder Unregelmäßigkeiten	Erfüllt
Benutzer-erfahrung	Sicherstellen einer benutzerfreundlichen Webseite	80% der Testteilnehmer bewerten die Webseite als benutzerfreundlich	60% der Testteilnehmer bewerten die Webseite als benutzerfreundlich	Teilweise erfüllt
Datenbank abfragen	Sicherstellen der Effizienz der Datenbank-abfragen	Abfragen innerhalb von 2 Sekunden abgeschlossen	Abfragen innerhalb von 2 Sekunden abgeschlossen, korrekte Ergebnisse	Erfüllt
Skalierbarkeit	Sicherstellen der effizienten Funktionalität bei steigender Nutzerzahl	Webseite unterstützt zusätzlich Benutzer ohne Leistungseinbußen	Lasttests erforderlich, Webseite unterstützt zusätzliche Benutzer	Erfüllt

Nicht funktionale Anforderung (Softwarequalität):

Ziel: Sicherstellen der Softwarequalität durch umfassende Tests.

Ergebnisse: die Software hat 99% der Testfälle bestanden und alle funktionalen sowie nicht funktionalen Anforderungen erfüllt.

Nicht Funktionale Anforderung (Funktionalität des Warenkorbs):

Ziel: Sicherstellen der korrekten Funktionalität des Warenkorbs.

Ergebnisse: alle Aktionen im Warenkorb werden korrekt ausgeführt, ohne Fehler oder Unregelmäßigkeiten.

Nicht funktionale Anforderung (Benutzererfahrung):

Ziel: Sicherstellen einer benutzerfreundlichen und intuitiven Webseite.

Ergebnisse: 60% der Testteilnehmer bewerteten die Webseite als benutzerfreundlich.

Nicht Funktionale Anforderung (Datenbankabfragen):

Ziel: Sicherstellen der Effizienz der Datenbankabfragen.

Ergebnisse: Datenbankabfragen werden innerhalb von maximal 2 Sekunden abgeschlossen und liefern korrekte Ergebnisse.

Nicht funktionale Anforderung (Skalierbarkeit):

Ziel: Sicherstellen der effizienten Funktionalität der Webseite bei steigender Nutzerzahl.

Ergebnisse: Die Webseite unterstützt zusätzliche Benutzer ohne Leistungseinbußen.

Bericht über die Umfrage zur Benutzerfreundlichkeit

Eine Umfrage zur Benutzerfreundlichkeit der Webseite wurde mit 15 Personen aus Wien im Alter von 19 bis 30 Jahren durchgeführt.

Ergebnisse der Umfrage:

- Teilnehmeranzahl: 15 Personen
- Altersgruppe: 19-30 Jahre
- Ort: Wien
- Bewertung der Benutzerfreundlichkeit: 60% der Befragten empfanden die Webseite als benutzerfreundlich.
- Fazit: obwohl die Mehrheit der Testteilnehmer die Webseite als benutzerfreundlich empfand, gibt es noch Raum für Verbesserungen. Das gesammelte Feedback wird verwendet, um die Benutzeroberfläche weiter zu optimieren und die Benutzerfreundlichkeit zu erhöhen.

Testprotokoll

Datum: 15.05.2024

Nach der Implementierung des Login- und Registrierungsformulars wurden verschiedene Tests durchgeführt, um die Funktionalität und Sicherheit dieser Komponenten zu überprüfen. Alle Tests wurden erfolgreich abgeschlossen, und keine Fehler wurden festgestellt.

Auch die Sicherheitsüberprüfungen zeigten, dass die Anwendung gegen gängige Angriffe gut geschützt ist. Testergebnis: Anwendung ist sicher gegen SQL-Injection und andere Angriffe.

Datum: 02.06.2024

Die Tests (inkl. Performance-Tests, Usability-Tests, Sicherheitsüberprüfungen) wurden durchgeführt, um die Funktionalität, Benutzerfreundlichkeit und Sicherheit der neuen Komponenten sicherzustellen. Es wurden auch die Tests mit verschiedenen Kombinationen von Eingabedaten durchgeführt, um mögliche Randfälle abzudecken.

Die durchgeführten Tests für die Navigationsleiste, die „About Us“-Seite und die Katalogseiten des Online-Shops haben bestätigt, dass alle Komponenten wie erwartet funktionieren. Die HTML-Strukturen, CSS-Stylings und JavaScript-Funktionen wurden erfolgreich implementiert und validiert. Die Integration von Backend-APIs ermöglicht eine dynamische und korrekte Darstellung der Produktdaten. Sicherheitsüberprüfungen ergaben keine Schwachstellen, und die Seiten sind responsive, was eine optimale Benutzererfahrung auf verschiedenen Geräten gewährleistet. Es wurden keine Fehler oder Probleme festgestellt, sodass die neuen Funktionen bereit für den Einsatz sind.

Datum 23.06.2024

Es wurden mehrere wesentliche Funktionen des Online-Shops erfolgreich geprüft:

- Kaufen-Funktion: die Funktion zum Hinzufügen von Artikeln zum Warenkorb und das Ändern der Menge;
- Logout-Prozess: der Logout-Button und die Sicherheit nach dem Logout;
- „Favorites“: das Hinzufügen und Entfernen von Artikeln zur Favoritenliste;
- Änderung der Menge/Farbe: die Funktionen zur Änderung der Menge und Farbe von Artikeln auf der Produktseite sowie deren Kombination;
- FAQ: die Anzeige der FAQ-Seite;
- Benutzerprofil: die Aktualisierung von Profilinformationen, das Einsehen der Bestellhistorie und das Ändern des Passworts;
- Einige Admin-Funktionen: neue Artikel hinzufügen, Bestellhistorie sowie auch die letzten Aktivitäten einsehen.

Alle getesteten Bereiche haben die Anforderungen vollständig erfüllt und zeigten keine Fehler.

Durchgeführte Tests:

Testfall-ID	Testbereich	Testfall	Erwartetes Ergebnis	Ergebnis
TF001	Login-Funktionalität	Einloggen mit gültigen Anmeldedaten	Benutzer wird erfolgreich	Erfolgreich

			authentifiziert und zur Startseite weitergeleitet	
TF002		Einloggen mit ungültigen Anmeldedaten (falsche E-Mail und/ oder Passwort)	Fehlermeldung wird korrekt angezeigt	Erfolgreich
TF003		Einloggen mit leerem E-Mail- oder Passwortfeld	Fehlermeldung wird korrekt angezeigt	Erfolgreich
TF004	Registrierungsfunktion	Registrierung mit gültigen Daten (Name, E-Mail, Passwort, Passwortbestätigung)	Benutzer wird erfolgreich registriert und zur Anmeldeseite weitergeleitet	Erfolgreich
TF005		Registrierung mit bereits verwendeter E-Mail-Adresse	Fehlermeldung wird korrekt angezeigt	Erfolgreich
TF006		Eingabe von nicht übereinstimmenden Passwörtern	Fehlermeldung wird korrekt angezeigt	Erfolgreich
TF007	Sicherheits-überprüfungen	Versuch, SQL-Injection-Angriffe durchzuführen	Anwendung wehrt solche Angriffe erforderlich ab	Erfolgreich
TF008		Überprüfung, ob Passwörter verschlüsselt in der Datenbank gespeichert werden	Passwörter werden korrekt verschlüsselt gespeichert	Erfolgreich
TF009	Navigationsleiste	HTML-Grundgerüst	Alle vorgesehenen Links und Menüpunkte sind vorhanden	Erfolgreich

TF010		CSS-Styling	Navigationsleiste ist optisch ansprechend und einheitlich	Erfolgreich
TF011		JavaScript-Dropdown-Funktion	Dropdown-Menüs öffnen und schließen sich korrekt bei Klick	Erfolgreich
TF012		Responsive Design	Navigationsleiste passt sich korrekt an verschiedene Bildschirmgrößen an	Erfolgreich
TF013		Sicherheitsprüfung	Keine XSS-Schwachstellen	Erfolgreich
TF014	„About Us“ - Seite	HTML-Inhalt- und Struktur	Inhalte werden vollständig und korrekt angezeigt	Erfolgreich
TF015		CSS-Styling	Seite ist optisch ansprechend und einheitlich	Erfolgreich
TF016		JavaScript-interaktive Elemente	Interaktive Elemente sind funktional und benutzerfreundlich	Erfolgreich
TF017		Integration von Bildern und Medien	Medieninhalte werden korrekt angezeigt und sind interaktiv	Erfolgreich
TF018		Responsive Design	Seite passt sich korrekt an verschiedene Bildschirmgrößen an	Erfolgreich

TF019		Sicherheitsprüfung	Keine XSS-Schwachstellen	Erfolgreich
TF020	Katalogseiten	HTML-Grundgerüst	Katalogseiten zeigen alle Produkte und Kategorien korrekt an	Erfolgreich
TF021		CSS-Styling	Katalogseiten sind optisch ansprechend und einheitlich	Erfolgreich
TF022		JavaScript-interaktive Elemente	Interaktive Elemente (Filter, Sortierfunktionen) sind funktional und benutzerfreundlich	Erfolgreich
TF023		Integration von Backend-APIs	Produktdaten werden korrekt angezeigt und aktualisiert	Erfolgreich
TF024		Responsive Design	Katalogseiten passen sich korrekt an verschiedene Bildschirmgrößen an	Erfolgreich
TF025		Sicherheitsprüfung	Keine XSS-Schwachstellen	Erfolgreich
TF026	Kaufen-Funktion (Warenkorb)	Artikel in den Warenkorb legen	Artikel wird korrekt zum Warenkorb hinzugefügt	Erfolgreich
TF027		Menge eines Artikels im Warenkorb ändern	Menge wird korrekt aktualisiert	Erfolgreich
TF028		Artikel aus dem Warenkorb entfernen	Artikel wird korrekt aus dem	Erfolgreich

			Warenkorb entfernt	
TF029	Logout-Prozess	Klicken auf den Logout-Button	Benutzer wird erfolgreich abgemeldet und zur Login-Seite weitergeleitet	Erfolgreich
TF030		Logout während eines Bestellvorgangs	Benutzer wird abgemeldet und der Bestellvorgang wird abgebrochen	Erfolgreich
TF031		Logout nach Passwortänderung	Benutzer wird nach der Passwortänderung automatisch abgemeldet und muss sich erneut einloggen	Erfolgreich
TF032	Favoriten	Hinzufügen eines Artikels zu den Favoriten	Artikel wird erfolgreich zur Favoritenleiste hinzugefügt	
TF033		Entfernen eines Artikels aus den Favoriten	Artikel wird erfolgreich aus der Favoritenleiste entfernt	
TF034		Überprüfung der Favoritenleiste nach dem Logout und erneuten Login	Favoritenleiste bleibt nach dem Logout und erneuten Login unverändert	
TF035	Die Änderung von Menge/ Farbe	Änderung der Menge eines Artikels auf der Produktleiste	Menge wird korrekt aktualisiert	Erfolgreich

TF036		Auswahl einer anderen Farbe des Artikels auf der Produktleiste	Farbänderung wird korrekt angezeigt	Erfolgreich
TF037		Kombination von Farb- und Mengenänderungen	Änderungen werden korrekt kombiniert und angezeigt	Erfolgreich
TF038	FAQ	Öffnen der FAQ-Seite	FAQ-Seite wird korrekt geladen und angezeigt	Erfolgreich
TF039		Suche nach einer Frage in der FAQ	Relevante Antworten werden korrekt angezeigt	Erfolgreich
TF040	Benutzerprofil	Aktualisierung/ Änderung von Profilinformationen	Änderungen werden korrekt und vollständig angezeigt	Erfolgreich
TF041		Einsicht in Bestellhistorie	Bestellhistorie wird korrekt und vollständig angezeigt	Erfolgreich
TF042		Änderung des Passworts	Passwort wird erfolgreich geändert und bestätigt	Erfolgreich
T043	Admin-Funktion: neue Artikel hinzufügen	Eingabe aller notwendigen Produktinformationen (Name, Beschreibung, ...)	Produkt wird erfolgreich hinzugefügt und ist im Shop sichtbar	Erfolgreich
T044		Hochladen von Produktbildern	Bildern werden erfolgreich	Erfolgreich

			hochgeladen und korrekt angezeigt	
T045		Speichern und Veröffentlichen des neuen Artikels	Artikel wird gespeichert und ist sofort im Shop verfügbar	Erfolgreich
T046	Admin-Funktion: Bestellhistorie einsehen	Anrufen der Bestellhistorie-Seite	Bestellhistorie-Seite wird korrekt geladen und angezeigt	Erfolgreich
T047		Ansicht der Detailinformationen einer Bestellung	Detaillierte Informationen zur Bestellung werden korrekt angezeigt	Erfolgreich
T049	Admin-Funktion: letzte Aktivitäten einsehen	Aufrufen der Seite mit den letzten Aktivitäten	Seite mit den letzten Aktivitäten wird korrekt geladen und angezeigt	Erfolgreich
T050		Anzeige der Detailinformationen zu einzelnen Aktivitäten	Detaillierte Informationen zur Aktivität werden korrekt angezeigt	Erfolgreich