

Inhaltsverzeichnis

1	Übersicht.....	4
2	Aufstellung	4
2.1	Ergebnisse Nicht Funktionale Anforderungen Frontend NFAFE	4
2.1.1	Responsives Webdesign [NFAFE ID-001]	4
2.1.2	Perfomance [NFAFE ID-002]	4
2.1.3	Verfügbarkeit [NFAFE ID-003].....	5
2.1.4	Nutzung von JWT [NFAFE ID-004].....	6
2.1.5	Formulare [NFAFE ID-005].....	6
2.2	Ergebnisse Funktionale Anforderungen Frontend FAFE.....	6
2.2.1	Navigation innerhalb der Webanwendung [FAFE ID-001]	6
2.2.2	Auflistung von Dienstleistungen [FAFE ID-002]	6
2.2.3	Volltextsuche & Filterung [FAFE ID-003]	6
2.2.4	Kontaktinformationen/öffentliches Profil [FAFE ID-004].....	7
2.2.5	Registrierung und Authentifizierung [FAFE ID-005]	7
2.2.6	Bereitstellung einer zentralen Übersicht. [FAFE ID-006].....	7
2.2.7	Verwaltung von Kontoeinstellungen und Profildaten. [FAFE ID-007]	7
2.2.8	Dienstleistungs-Verwaltung [FAFE ID-008]	7
2.2.9	Öffentliches Profil [FAFE ID-009]	7
2.2.10	Passwort zurückzusetzen [FAFE ID-010]	7
2.2.11	Kontaktformular [FAFE ID-011]	8
2.2.12	Marketinganalysen [FAFE ID-012].....	8
2.2.13	Usability [FAFE ID-013].....	8
2.3	Ergebnisse Nicht Funktionale Anforderungen Backend NFABE	8
2.3.1	Leistung [NFABE ID-001].....	8
2.3.2	Sicherheit [NFABE ID-002].....	9
2.4	Ergebnisse Funktionale Anforderungen Backend FABE.....	9
2.4.1	Request-Verarbeitung [FABE ID-001]	9
2.4.2	Datenverarbeitung und Konsistenz [FABE ID-002]	9
2.4.3	Authentifizierung [FABE ID-003]	9
2.4.4	Registrierung [FABE ID-004]	10
2.4.5	Kontoverwaltung (CRUD) [FABE ID-005]	10
2.4.6	Abrufen aller verfügbaren Dienstleistungen. [FABE ID-006]	10
2.4.7	Verwaltung von Dienstleistungs – Daten [FABE ID-007].....	10
2.4.8	Öffentliches Profil [FABE ID-008].....	10
2.4.9	Passwort zurücksetzen [FABE ID-009]	10

2.4.10	Benachrichtigung [FABE ID-010]	11
2.4.11	Analyse für Marketingzwecke [FABE ID-011]	11
3	Fazit	11

1 Übersicht

Gemäss Modulunterricht, soll das Projekt in einen finalen Stand gebracht werden. Zur Auswertung soll eine Dokumentation vorhanden sein welche eine Auswertung der Semesterarbeit aufzeigt, besonders sollen die erzielten Ergebnisse gemäss definierten Anforderungen aufgezeigt werden.

2 Aufstellung

Die Ergebnisse werden wie in der Dokumentation der Semesterarbeit, in Funktionalen und Nichtfunktionalen Anforderungen gegliedert. Die Definition der Anforderung ist im Dokument Spezifizierung_technische_Anforderungen enthalten.

2.1 Ergebnisse Nicht Funktionale Anforderungen Frontend NFAFE

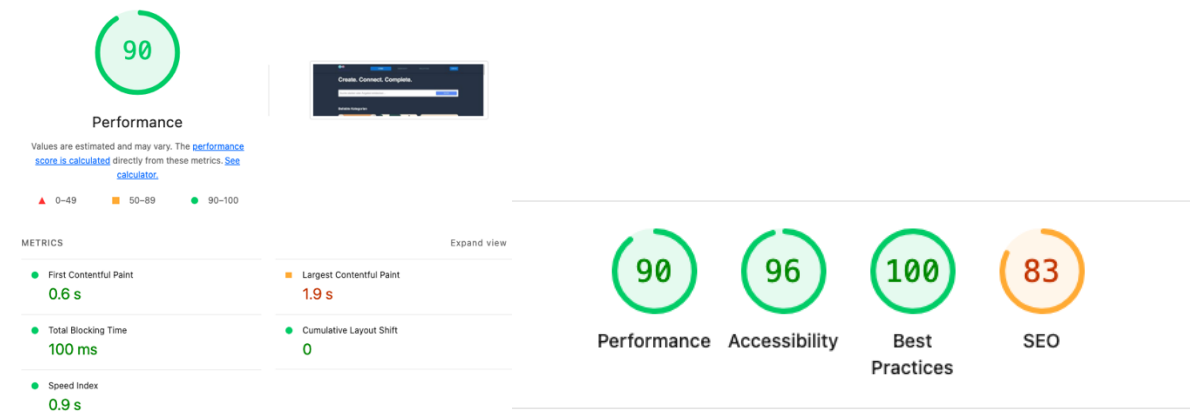
2.1.1 Responsives Webdesign [NFAFE ID-001]

Die Applikation wurde mit Chrome und Firefox geprüft und ist auf allen Bildschirmgrössen bedienbar.

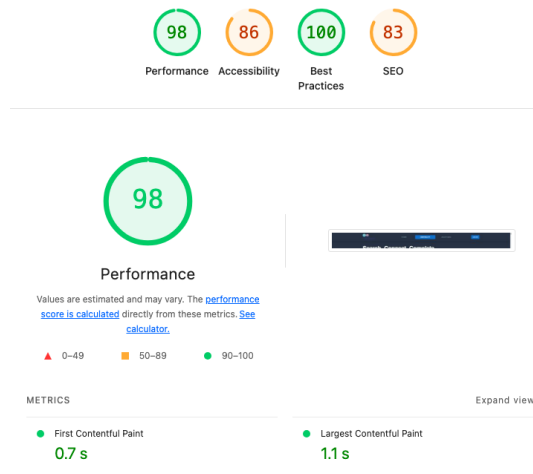
2.1.2 Perfomance [NFAFE ID-002]

Die Applikation gewährleistet eine akzeptable Performance. Ich habe die wichtigsten Aufrufe der Applikation in einer lokalen Produktionsumgebung mit Google Lighthouse geprüft.

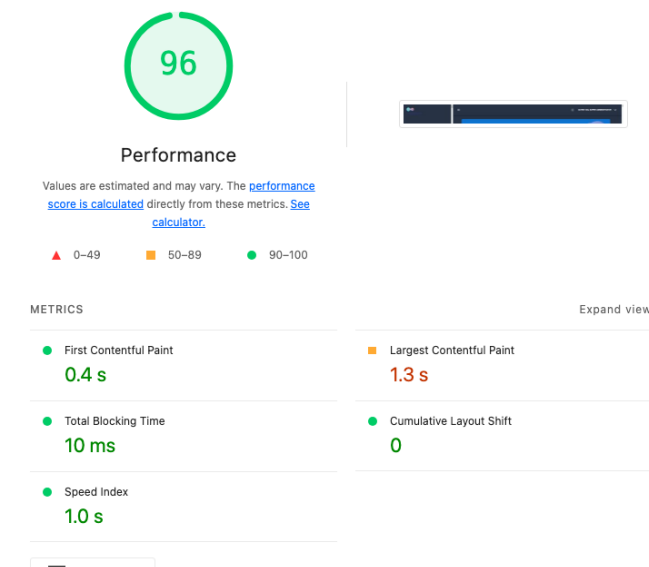
Startseite:



Board (Übersicht aller Dienstleistungen):



Aufruf Dashboard (registrierter Benutzer):



Die Auswertung zeigt, dass die Anforderung erfüllt wurde.
Für die Prüfung wurde ein Setup verwendet mit ca. 25 Registrierten Services.

Die Performance während der Entwicklung (mit dev-server) ist initial nicht optimal – für die Lesbarkeit und Wartbarkeit des Codes wurde ein Setup verwendet das bei Bedarf angepasst werden kann, dies betrifft hauptsächlich die Imports der MUI – Komponenten. Dies ist allerdings nicht relevant sobald ein production - bundle der Applikation verwendet wird.

2.1.3 Verfügbarkeit [NFAFE ID-003]

Die Applikation gewährleistet eine konsistente Nutzererfahrung. Es werden alle möglichen/bekannte Fehlerfälle und Ereignisse zuverlässig abgedeckt. Um diese Anforderung zu erfüllen wurde ein Error Handling Interface entwickelt welche dem Nutzer detaillierte Auskunft zum Fehler bietet und Lösungsmöglichkeiten vorschlägt.

Jenach Use-Case (Informations-, Fehlermeldung) wird der Nutzer per Modal/Dialog informiert bezüglich des Stands der Applikation.

Fehlerhafte Anfragen werden auf eine 404 – Seite umgeleitet, falls versucht wird auf Seiten zu navigieren welche eine Authentifizierung benötigen wird bei fehlenden Rechten auf die Startseite umgeleitet.

2.1.4 Nutzung von JWT [NFAFE ID-004]

Die Applikation verwendet durchgehend JWT für Authentifizierungsabläufe und das ausgestellte Token (Access) bei Sicherheitsrelevante Anfragen.
Ist das Token (Refresh) abgelaufen wird der Nutzer zur erneuten Anmeldung aufgefordert. Und eine weitere Programmausführung gestoppt. Im Falle eines Logouts wird das Token serverseitig invalidiert und vom Client entfernt.

Abweichungen zum definierten Standartablauf: Das Token wird im Local Storage gespeichert nicht im Session Storage – damit der Nutzer automatisch eingeloggt wird beim erneuten öffnen der Applikation.

2.1.5 Formulare [NFAFE ID-005]

Formulareingaben werden zuverlässig Validiert und geprüft, auch ein Honeypot- Feld wurde implementiert. Die Applikation verwendet definierte Validierungsschemas diese wurden hier umgesetzt: *Sidelink/Frontend/src/config/Schemas.js*

Durchgehend alle Formulare der Applikation werden validiert, ich werde in weiteren Anforderungsauswertungen nicht nochmals drauf eingehen.

2.2 Ergebnisse Funktionale Anforderungen Frontend FAFE

2.2.1 Navigation innerhalb der Webanwendung [FAFE ID-001]

Eine konsistente Navigation der Anwendung wurde mit react-router implementiert. Und ist auf allen Bildschirmgrößen verwendbar.

2.2.2 Auflistung von Dienstleistungen [FAFE ID-002]

Die zentrale Übersicht ist auf */board* verfügbar. Die zentrale Übersicht ist öffentlich Verfügbar und bietet eine Ansicht der erstellten Angebote. Für eine detaillierte Ansicht eines Angebots (Suche/Biete) ist eine Anmeldung auf der Applikation notwendig. Die Ergebnisse werden tabellarisch dargestellt und es können vom Nutzer Einstellungen gemacht werden für die Paginierung innerhalb der Ergebnisse/Suchresultaten.

2.2.3 Volltextsuche & Filterung [FAFE ID-003]

Die Such- und Filterfunktionen sind ebenfalls auf der Seite */board* implementiert. Der Nutzer wird informiert falls keine Suchergebnisse gefunden wurden. Die Applikation bietet eine akzeptable Performance für bis zu 2000 Dienstleistungsangebote.

Details der Umsetzung:

- Die Volltextsuche wird basierend auf den aktiven Filtern durchgeführt.
- Die Filter sind UND Verknüpft mit Ausnahme der Subkategorien – ein Service wird in die Ergebnisse aufgenommen, wenn eine der ausgewählten Subkategorien vorhanden ist.
- Wenn ein Filter neu gesetzt wird, wird die Volltextsuche zurückgesetzt.

2.2.4 Kontaktinformationen/öffentliches Profil [FAFE ID-004]

Um Einsicht auf ein öffentliches Profil zu erhalten – muss ein Benutzer registriert sein. Ein Angebot kann ausgewählt werden und man wird auf eine entsprechende Seite weitergeleitet wo das öffentliche Profil ersichtlich ist mitsamt den Details des Angebots.

Die Pflege des öffentlichen Profils wird dem Benutzer überlassen, kann also auch leer sein. Falls Missbrauch auftreten sollte, werden die AGB über weitere Verläufe entscheiden.

2.2.5 Registrierung und Authentifizierung [FAFE ID-005]

Ein Nutzer kann sich auf der Seite */Registration* anmelden und nach erfolgreicher Anmeldung auf der Seite */login* einloggen und die Funktionen der Applikation nutzen.

2.2.6 Bereitstellung einer zentralen Übersicht. [FAFE ID-006]

Der angemeldete Nutzer kann über das Menü auf sein Dashboard zugreifen. Innerhalb des Dashboards hat der Nutzer eine weitere Navigation für die Navigation innerhalb des Dashboards.

2.2.7 Verwaltung von Kontoeinstellungen und Profildaten. [FAFE ID-007]

Der angemeldete Benutzer kann durch Auswahl *Account Verwaltung* innerhalb des Dashboard - Menüs seine Accountdaten einsehen und bei Bedarf ändern. Zudem kann er auch seinen Account löschen. Änderungen werden nach dem Speichern umgehend vom System übernommen und angezeigt, falls das Konto gelöscht wird erhält der Nutzer eine Informationsmeldung für 3 Sekunden und wird dann automatisch ausgeloggt und zur Startseite weitergeleitet.

2.2.8 Dienstleistungs-Verwaltung [FAFE ID-008]

Der angemeldete Benutzer kann durch Auswahl *Meine Angebote / Meine Suche* innerhalb des Dashboard - Menüs alle CRUD-Operationen bezüglich Angebote durchführen. Änderungen werden nach dem Speichern umgehend vom System übernommen und angezeigt.

Die optionale Kann – Anforderung: Suche nach Dienstleistungen wurde ebenfalls umgesetzt und hat identische Funktionalität zur Erstellung von Angeboten. (*Meine Suche*)

2.2.9 Öffentliches Profil [FAFE ID-009]

Der angemeldete Benutzer kann durch Auswahl *Öffentliches Profil* innerhalb des Dashboards -Menü sein öffentliches Profil bearbeiten. Änderungen werden nach dem Speichern umgehend vom System übernommen und angezeigt.

2.2.10 Passwort zurückzusetzen [FAFE ID-010]

Ein Benutzer der Applikation kann eine Anfrage zum Zurücksetzen des Passworts anfordern. Dies kann er tun im Bereich des Logins (*Passwort vergessen?*). Falls die angegebene E-Mail im System registriert ist wird eine E-Mail an die entsprechende Adresse versendet.

Die versendete E-Mail enthält einen Link welcher benutzt werden kann um das Passwort zu ändern (3 Tage Gültigkeit)

Beispiel Link:

<http://localhost:5173/password-reset/MQ/crgba3-78c925e1b0d432d7a230af95229eb563/>

Die Applikation ist in der Testumgebung so konfiguriert, dass der Link in der Django Konsole ausgegeben wird.

2.2.11 Kontaktformular [FAFE ID-011]

Ein Benutzer soll die Möglichkeit haben die Betreibenden der Applikation zu kontaktieren. Dies wurde mit einem Kontaktformular unter **/contact** umgesetzt. Falls eine Anfrage gemacht wird, wird die E-Mail an eine mittels .env definierte E-Mail-Adresse versandt.

Die Applikation ist in der Testumgebung so konfiguriert, dass die E-Mail in der Django Konsole ausgegeben wird.

2.2.12 Marketinganalysen [FAFE ID-012]

Gemäss Anforderung werden die definierten Analysen im Frontend dargestellt. Ein registrierter Benutzer hat die Möglichkeit innerhalb des Dashboards auf die Seite **/dashboard/analytics** zu navigieren und dort die Analysen einzusehen.

Abweichung zur Spezifikation der Anforderung:

Es wurden zur Ansicht der Analysen nur Superadministratoren definiert. Von dieser Definition wurde abgewichen und die Analysen sind für registrierte Benutzer einsehbar, damit soll die Verwendung der Applikation durch Benutzer optimiert werden, oder ein User soll die Möglichkeit haben optimierte Dienstleistungen zu erstellen.

Die definierte Kann Anforderungen bezüglich der grafischen Darstellung wurde umgesetzt.

2.2.13 Usability [FAFE ID-013]

Eine benutzende Person kann die auf der Applikation erstellten Such- und Filterparameter erfolgreich via URL speichern oder teilen.

Auch die Detailansicht eines Inserats (öffentliches Profil mit Details des ausgewählten Inserats) kann gespeichert und geteilt werden, Voraussetzung dafür ist das der Benutzer im System angemeldet ist.

2.3 Ergebnisse Nicht Funktionale Anforderungen Backend NFABE

2.3.1 Leistung [NFABE ID-001]

Das Backend bietet eine akzeptable Performance und kann einwandfrei in Betrieb genommen werden. Die Auswertung der Performance wurde mit der Library Silk beurteilt. Es bestehen Optimierungsmöglichkeiten im Bereich des Logins und der Datenaufbereitung für das Dashboard. Primäre Priorität im Prüfen der Leistungsfähigkeit hat der öffentliche Bereich der Applikation.

Die Applikation ist in der Testumgebung so konfiguriert, dass im Development Modus auf Silk zugegriffen werden kann:

<http://127.0.0.1:8000/silk/>

2.3.2 Sicherheit [NFABE ID-002]

Das Backend enthält diverse Sicherheitsmechanismen um eine konsistente Sicherheit zu gewährleisten.

- Die Applikation nutzt das integrierte Django-Authentifizierungssystem für den Zugriff auf die Verwaltungsoberfläche (Admin). Zur Erhöhung der Sicherheit wurden zusätzlich eigene Passwort – Validatoren implementiert.
- Passwörter werden als sensible Daten klassifiziert und werden niemals im Klartext gespeichert. Sie werden durch das Django interne Hash-Verfahren in der Datenbank abgelegt.
- Für den Zugriff auf Benutzerdaten ist im Backend das Sicherheitsverfahren mit JWT implementiert, jede Anfrage bezüglich solcher Ressourcen erfordern ein gültiges Token.
- Alle im Backend eingehende Anfragen an das System werden geprüft und ggfs. validiert und bereinigt.
- Im Fehlerfall wird die eingehende Anfrage abgelehnt mit einer entsprechenden Status Meldung

2.4 Ergebnisse Funktionale Anforderungen Backend FABE

Entsprechend der Anforderungen implementiert das System eine REST-API, deren Endpunkte mittels ‘drf_spectacular’ dokumentiert wurden. Die API-Dokumentation ist nach der Inbetriebnahme des Systems über die Root-URL (<http://127.0.0.1:8000/>) einsehbar.

2.4.1 Request-Verarbeitung [FABE ID-001]

Das System stellt eine Funktionsfähige REST-API zur Verfügung. Es können gemäss definierten Endpunkten CRUD Operationen durchgeführt werden.

2.4.2 Datenverarbeitung und Konsistenz [FABE ID-002]

Das System ist in der Testumgebung mit einer PostgreSQL Datenbank verbunden. Mittels der API-Endpunkten werden Daten strukturiert verarbeitet, validiert und gemäss Django-ORM-Modellen in der Datenbank abgelegt. Daten werden effizient gemäss ORM-Modellen zur Aufbereitung oder direkten Auslieferung and die Endpunkte zur Verfügung gestellt.

Für eine konsistente Datenhaltung wurde jedem ORM-Modell eine Timestamp-Mixin hinzugefügt. (automatische Felder für Datumsangaben bezüglich Neuerstellung oder Änderungen)

2.4.3 Authentifizierung [FABE ID-003]

Das System verwendet zur Authentifizierung den JWT-Standardablauf. Es werden die üblichen Endpunkte zur Anfrage, Invalidierung und Erneuerung von Tokens zur Verfügung gestellt.

2.4.4 Registrierung [FABE ID-004]

Das System stellt einen Endpunkt zur Verfügung bezüglich der Registrierung von Nutzern. Für die Registrierung wurde das Django interne Registrierungsverfahren gemäss Anforderungen der Applikation modifiziert. Standardmässig erhält eine neue Registration keine weiteren Rechte und wird als aktiv gekennzeichnet. Es besteht keine weiteren Spezifikationen für Gruppen oder Rechte (abgesehen vom Erstellen eines Superadmins per Konsole)

Standardmässig wird bei einer neuen Registration ein öffentliches Profil erstellt und der neuen Registration zugewiesen.

2.4.5 Kontoverwaltung (CRUD) [FABE ID-005]

Das System stellt API-Endpunkte zur Verfügung um alle CRUD-Operationen am registrierten Konto durchzuführen. Es können alle registrierten Daten bearbeitet werden.

Falls das Konto gelöscht werden soll, wird keine Passwort Abfrage gemacht, da bei einer solchen Anfrage ein gültiges Token Voraussetzung ist.

2.4.6 Abrufen aller verfügbaren Dienstleistungen. [FABE ID-006]

Das System stellt einen öffentlichen API-Endpunkt zur Verfügung welche alle verfügbaren Inserate (Angebot/Suche) im Backend aufbereitet und zur Verfügung stellt.

2.4.7 Verwaltung von Dienstleistungs – Daten [FABE ID-007]

Das System stellt API-Endpunkte zur Verfügung um dem authentifizierten (mittels JWT-Tokens) Benutzer volle CRUD-Operationen auf Dienstleistung Angebote zu ermöglichen.

Die Kann Anforderung für das Erstellen von Inseraten (Suche nach Dienstleistung) wurde ebenfalls implementiert und erfolgreich umgesetzt.

2.4.8 Öffentliches Profil [FABE ID-008]

Das System stellt API-Endpunkte zur Verfügung um dem authentifizierten (mittels JWT-Tokens) Benutzer die Möglichkeit zu bieten sein öffentliches Profil zu bearbeiten. Die Funktionalität zum Löschen des öffentlichen Profils wurde nicht implementiert da es Voraussetzung für die erfolgreiche Nutzung der Applikation ist.

Ein öffentliches Profil – implementiert ein Profilbild – diese Bilder werden im media Ordner des Projekts abgelegt und eine Referenz in der Datenbank gespeichert. Bilder werden ausführlich validiert bevor sie persistiert werden.

2.4.9 Passwort zurücksetzen [FABE ID-009]

Das System stellt öffentliche API-Endpunkte zur Verfügung um das dem Benutzer zugewiesen Passwort zurückzusetzen. Dies kann der Benutzer via Dashboard machen oder mittels Anfrage an den entsprechenden Endpunkt mit seiner E-Mail-Adresse. Das System generiert mittels der E-Mail-Adresse (falls im System vorhanden) einen Link mit Token und UID. Mittels der implementierten Businesslogik wird die E-Mail versendet.

Falls der Link verwendet wird, wird im entsprechenden Endpunkt das Token und die UID zur Authentifizierung verwendet. Bei erfolgreicher Authentifizierung wird das Passwort geändert.

In der Testumgebung wird die generierte E-Mail direkt in der Django Konsole ausgegeben.

2.4.10 Benachrichtigung [FABE ID-010]

Das System kann Benachrichtigungen in Form von Emails erstellen und diese versenden.

Zusätzlich wird jede Anfrage (per Kontaktformular) in der Datenbank persistiert.

Um mögliche Fehleranalysen durchzuführen wurde ein E-Mail-Logger implementiert welcher den Versand von Emails oder eventuelle Fehler loggt.

2.4.11 Analyse für Marketingzwecke [FABE ID-011]

Die Analysen wurden gemäss Spezifikationen im System implementiert. Unter der Annahme das, dass System weiterentwickelt wird und komplexere Auswertungen/Statistiken erstellt werden können wurde dazu eine eigene Django-App erstellt. Mittels Businesslogik werden die Daten aufbereitet und dem Endpunkt zur Verfügung bereitgestellt.

Das System berechnet die Conversation-Rate und stellt sie zur Verfügung. Dazu wurde eine eigenes Django-ORM-Model erstellt welches die Seitenaufrufe persistiert.

3 Fazit

Ich konnte alle definierten MUSS-Anforderungen erfolgreich umsetzen. Auch konnte ich weitere optionale KANN Anforderungen umsetzen. Ich persönlich bin zufrieden mit den Ergebnissen, und freue mich darauf dieses Projekt weiterzuführen.