



AKTIEN WEBSITE UND AKTIEN TELEGRAM BOT

LINUS EICKHOFF, FLORIAN KELLERMANN, FLORIAN KAISER, KEVIN PAUER - TEAM 1 (WEB ENGINEERING 2)

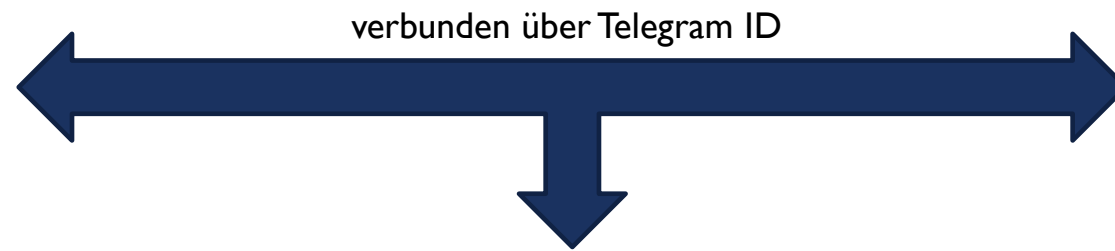
GLIEDERUNG

- Produktumfang
- Team Overview
- Sprachen und Frameworks
- Telegram Bot
- Website
- Backend
- Showcase
- Lessons learned und Tipps

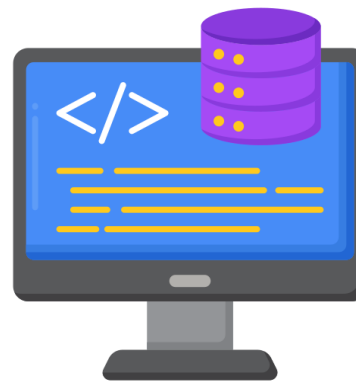
PRODUKTUMFANG



Website für Aktien- und Accountverwaltung



Telegram-Bot für Aktien und News



Backend

TEAM OVERVIEW

Name und Github	Email und Matr. Nr.	Aufgaben
Linus Eickhoff (Ripped)	wi20034@lehre.dhbw-stuttgart.de (1943478)	Projektstruktur, Datenbankdesign, Dokumentation, Telegram Bot, API-Handler, News-fetching
Florian Kellermann (NormalParameter)	inf20141@lehre.dhbw-stuttgart.de (8838597)	Projektstruktur, Datenbankdesign, Dokumentation, Telegram Bot, Update-Scheduler, Stocks-fetching
Florian Kaiser (H4CK3R-01)	inf20155@lehre.dhbw-stuttgart.de (9829423)	Projektstruktur, Datenbankdesign, Dokumentation, Backend (Datenbank, API, Server, CI-CD)
Kevin Pauer (kevinpauer)	inf20003@lehre.dhbw-stuttgart.de (1199719)	Projektstruktur, Datenbankdesign, Dokumentation, Frontend (Website)

SPRACHEN UND FRAMEWORKS

Telegram Bot und Backend

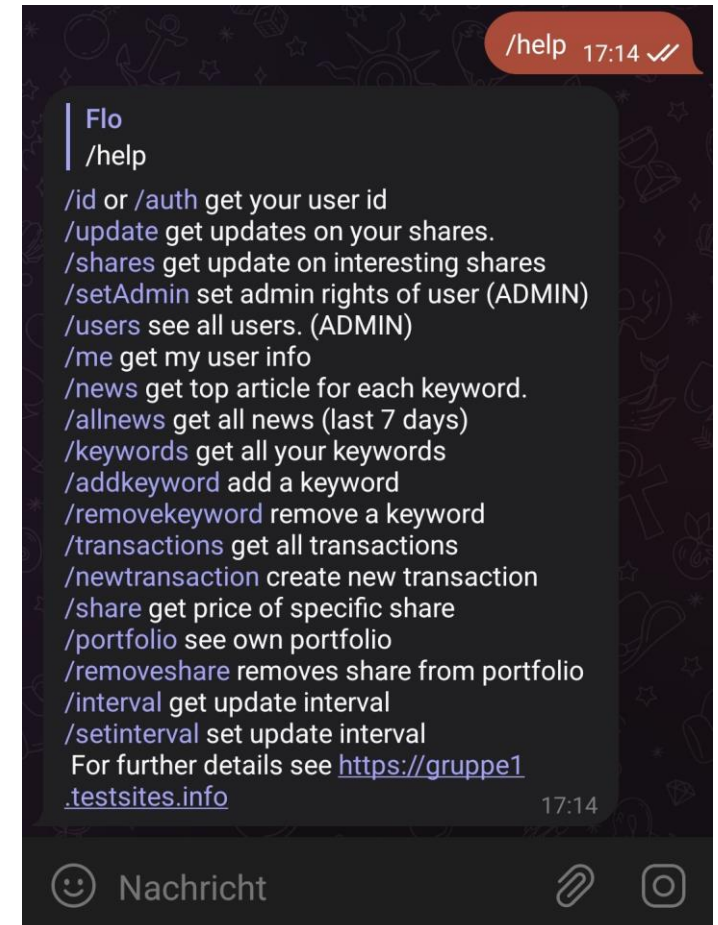


Website



TELEGRAM BOT

- Erreichbar unter @projektaktienbot
- Erweiterung der Funktionen der Website
- /setinterval zum ändern oder setzen des intervals in dem man updates erhalten möchte
- Täglich updates beinhalten:
 - Aktien die man besitzt
 - Aktien an denen man interessiert ist
 - Nachrichten zu seinen Stichwörtern

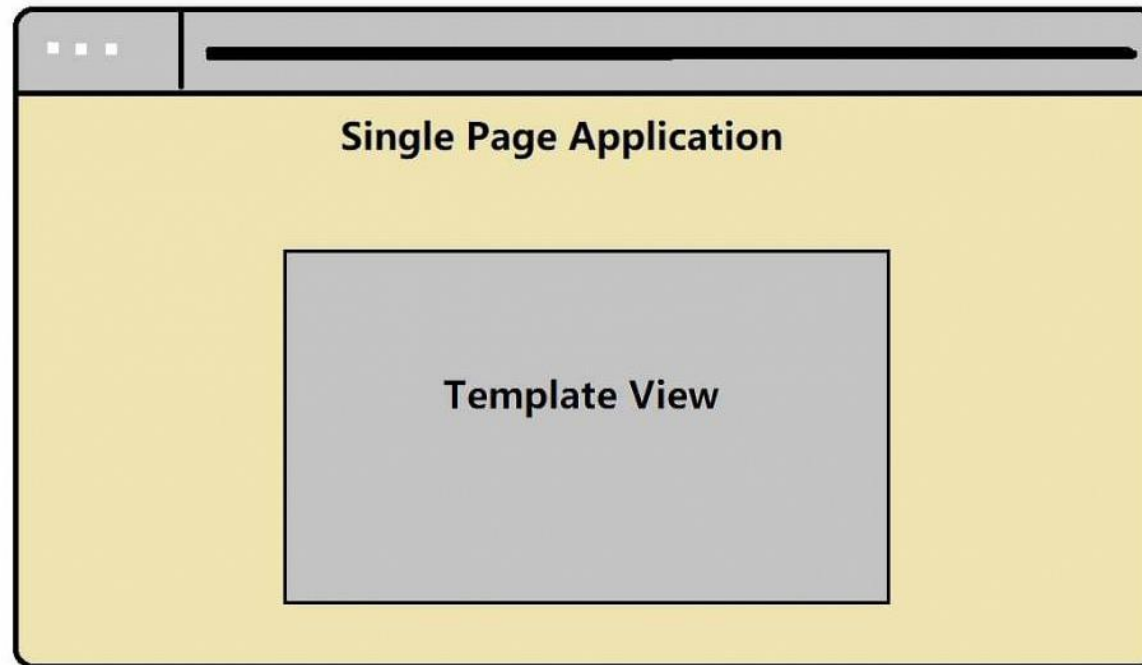


WEBSITE

Template 1

Template 2

Template 3



BACKEND & DATENBANK INFRASTRUKTUR

- APIFlask
- Automatische Swagger Dokumentation
- Passwort mithilfe von bcrypt + Salt gespeichert
- Skript, um aktuellen Kurs zu speichern wird mit Cron-Job regelmäßig gestartet
- Nginx Webserver mit UWSGI
- Docker Healthcheck

curl -s http://localhost:80/ -o /dev/null || exit 1

share_price	
PK: id	int(11)
isin	varchar(255)
price	float
date	datetime

Database V0.1

- Docker + Docker-Compose

- Watchtower
- Traefik
- MariaDB + phpMyAdmin
- Bot, Frontend, API

shares	
PK: a_id	int(11)
FK: email	varchar(255)
Admin	varchar(255)
comment	varchar(255)

users	
PK: email	varchar(255)
password	varchar(255)
username	varchar(255)
telegram_user_id	varchar(255)
admin	tinyint(1)
cron	varchar(20)

- Woodpecker (CI-CD)

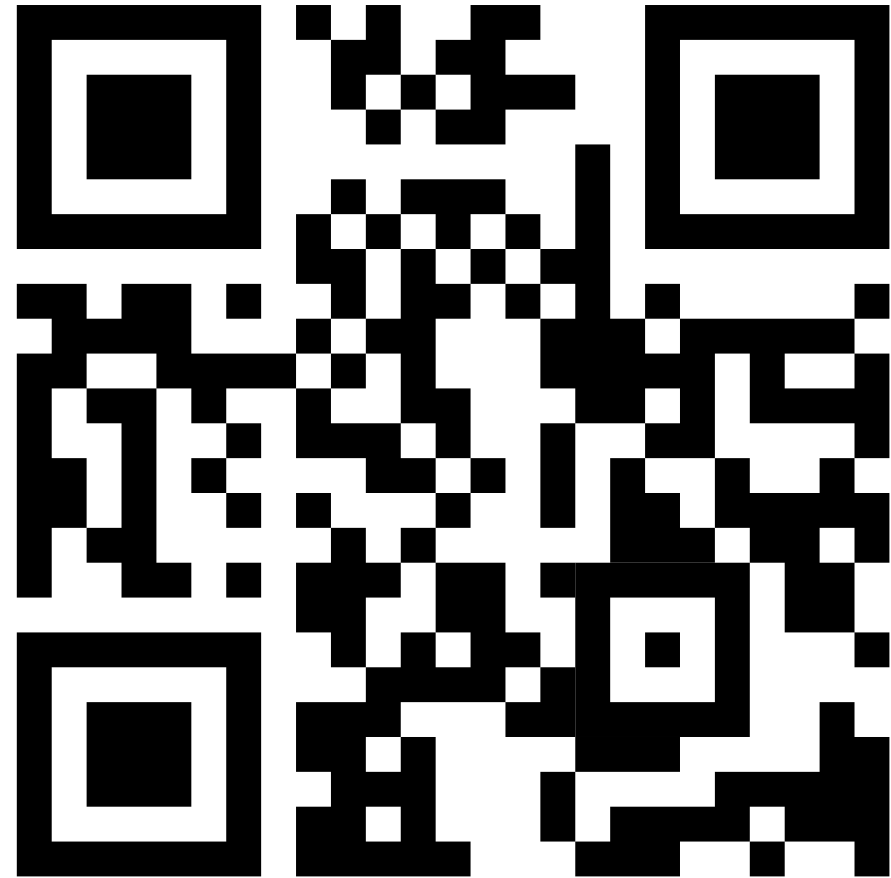
- Uptimekuma
- Dependabot

keywords	
PK: s_id	int(11)
FK: email	varchar(255)
keyword	varchar(255)

LIVE SHOW CASE

[HTTPS://GRUPPE1.TESTSITES.INFO/](https://GRUPPE1.TESTSITES.INFO/)

[HTTPS://T.ME/PROJEKTAKTIEBOT](https://t.me/projektaktienbot)



FAZIT UND EMPFEHLUNGEN

■ **Lessons learned**

- Höherer Fokus auf Usability als auf Funktionsumfang
- Gute API's kosten was
- Inputs validieren
- Markdown in Nachrichten riskant
- Frühzeitig anfangen ausführlich zu testen

■ **Empfehlungen**

- Compodoc und Dependabot für Doku und Kontinualität
- Crontab für Automatisierung
- Automatische Container Builds



VIELE DANK

[HTTPS://GRUPPE1.TESTSITES.INFO/](https://GRUPPE1.TESTSITES.INFO/)

[HTTPS://T.ME/PROJEKTAKTIEBOT](https://t.me/projektaktienbot)

