

Schulische Projektarbeit Sommer 2022

Fachinformatiker für Anwendungsentwicklung
Dokumentation zur schulischen Projektarbeit

Projekt „Kraftfahrzeugkennzeichen“

Webapplikation zur Informationsausgabe anhand des
Unterscheidungszeichen des KFZ-Kennzeichens

Abgabetermin: Darmstadt, den 20.05.2022

Projektbeteiligte:

Piotr Dziekoński
Aldi Wijaya Hutani
Viet Duy Peter Pham



Ausbildungsbetrieb:

Heinrich-Emanuel-Merck-Schule
Alsfelder Straße 23
64289 Darmstadt

Inhaltsverzeichnis

Abbildungsverzeichnis	II
Tabellenverzeichnis	III
Listings	IV
Abkürzungsverzeichnis	V
1 Ausgangssituation	1
1.1 Projektziel und Teilaufgaben	1
1.2 Projektumfeld, Prozessschnittstellen	1
2 Ressourcen- und Ablaufplanung	1
2.1 Personal-, Sachmittel-, Termin- und Kostenplanung	1
2.2 Ablaufplanung	2
3 Durchführung und Auftragsbearbeitung	2
3.1 Prozessschritte und Vorgehensweise	2
3.2 Qualitätssicherung	3
3.3 Abweichungen & Anpassungen	3
4 Projektergebnisse	4
4.1 Soll-/Ist-Vergleich	4
4.2 Lesson Learned	4
A Anhang	i
A.1 Use-Case-Diagramm	i
A.2 Anwendungsfall-Beschreibung	i
A.3 GANTT-Diagramm	iv
A.4 Einpflegen der Datenbank-Tabelle	v
A.5 Skript	viii
A.6 Datenbank Anbindung mit Express-Back-End	ix
A.7 Abrufen der SQL-Abfrage in Back-End Anwendung	ix
A.8 Benutzerdokumentation	xiv

Abbildungsverzeichnis

1	Use-Case Diagramm für die Anwendung KFZ-Kennzeichnung	i
2	GANTT-Diagramm zur Visualisierung der Schritte zum Projektziel	iv
3	Das Laden der JSON Datei in die neue Tabelle	v
4	Konfigurierung der Tabelle	v
5	Beginn des Importierens	vi
6	Beginn des Importierens	vi
7	Ansicht der Tabelle KFZ Kennzeichnung	vi
8	Ansicht der Datenbank-Inhalt in Back-End-Anwendung	vii
9	Skript zum Extrahieren der KFZ-Kennzeichen Daten	viii
10	Die Ansicht der Anwendung des KFZ-Verwaltungsystems	xiv

Tabellenverzeichnis

1	User Flow's	3
2	Soll-/Ist-Vergleich	4

Listings

- | | | |
|---|---|----|
| 1 | Javascript Code für die Herstellung der Verbindung mit Datenbank | ix |
| 2 | Javascript Code für bestimmten Datensatz aus der Tabelle abzufragen | ix |

Abkürzungsverzeichnis

CSS	Cascading Style Sheets
CSV	Comma Separated Value
SQL	Structured Query Language
HEMS	Heinrich Emanuel Merck Schule
KFZ	Kraftfahrzeug
XML	Extensible Markup Language
JSON	Javascript Object Notation

1 Ausgangssituation

1.1 Projektziel und Teilaufgaben

Das Ziel dieses Projektes ist die erfolgreiche Entwicklung einer webbasierten Anwendungen, zur einfachen Informationsausgabe eines Autokennzeichens. Anhand des Anfangsbuchstaben des Kennzeichens sollen Kreis und Bundesland ausgegeben werden. Zusätzlich sollen Kennzeichen nach Bundesländern, Kreisen und Kreisstädten ermittelt werden können. Durch die Einbindung von Wikipedia und Google Maps soll der Benutzer die Möglichkeit haben, weitere Informationen bei Bedarf zu erhalten. Des Weiteren sollen für den Datenaustausch die Möglichkeit zum Import und Export bestehen. Zu diesen Anforderungen wurde ein Use-Case-Diagramm erstellt, welches unter A.1 - Use-Case-Diagramm zu finden ist. Zusätzlich zu dem Use-Case-Diagramm ist eine genauere Beschreibung der Anwendungsfälle in tabellarischer Form im Anhang unter A.2 - Anwendungsfall-Beschreibung zu finden.

Der Auftraggeber wünscht sich eine beliebige Funktion, welche die Autoren frei wählen dürfen. Auf diesen Wunsch wird im Abschnitt 3.3 - Abweichungen, Anpassungen, Entscheidung näher eingegangen.

1.2 Projektumfeld, Prozessschnittstellen

Die Heinrich-Emanuel-Merck-Schule Darmstadt ist eine selbstständige berufliche Schule für Elektro- und Informationstechnik. Im Bereich IT bietet die Heinrich Emanuel Merck Schule (HEMS) Vertiefung im Bereich Anwendungsentwicklung, System-Integration und kaufmännischen Unterricht an. Unsere Gruppe mit der Vertiefung in der Anwendungsentwicklung besteht aus den im Deckblatt genannten Autoren Aldi Hutani, Piotr Dziekonski, Peter Pham, hat sich mit dem Projekt Kraftfahrzeugkennzeichen-Verwaltung befasst. Das Projekt mit dem Namen "Kraftfahrzeugkennzeichen" wird in den Räumlichkeiten der HEMS durchgeführt. Hauptansprechpartner ist Herr Norbert Grothe, der dieses Projekt genehmigt hat und den Autoren alle Mittel für die Umsetzung des Projektes zur Verfügung stellt.

2 Ressourcen- und Ablaufplanung

2.1 Personal-, Sachmittel-, Termin- und Kostenplanung

Zur Entwicklung dieses Projektes waren drei Auszubildende mit der Vertiefung Anwendungsentwicklung eingesetzt. Durch den Projektstart am 09.02. und der Abgabe am 20.05.2022 haben die Autoren einen Zeitraum von 5 Schulwochen für die erfolgreiche Bearbeitung des Projektes. Jeden Mittwoch eines Schulblockes wird den Autoren von 08:00 bis 13:00 Uhr die Möglichkeit gegeben, an dem Projekt zu arbeiten. Dadurch ergibt sich ein theoretischer Arbeitsaufwand von 25 Stunden. Dieser Arbeitsaufwand wird nicht durch den Auftraggeber bezahlt sondern durch die auszubildenden Firmen. Dadurch entstehen Projektkosten in Höhe von null Euro für den Auftraggeber.

2.2 Ablaufplanung

Zur Erreichung des Projektziels wurden Teilaufgaben erstellt und diese innerhalb der Projektgruppe aufgeteilt. Diese zu bearbeiteten Teilaufgaben wurden zur besseren Übersicht in einem GANTT Diagramm festgehalten. Dieses Diagramm ist im Anhang A.3: GANTT-Diagramm auf Seite iv einzusehen.

3 Durchführung und Auftragsbearbeitung

3.1 Prozessschritte und Vorgehensweise

Zu aller erst wurde eine Datenbank benötigt. Um die Datenbank abzurufen, wird die Programmiersprache Structured Query Language (SQL) angewendet. Wobei MySQL zum Einsatz kommt, um die Daten-Speicherung, -Verwaltung, und -Änderung in die Datenbank vorzunehmen.

Hierfür wurde Folgendes durchgeführt. Im Anhang unter Anhang A.4 sind Ausschnitte dazu zu finden.

- Installation sowie Einrichtung der Datenbank auf lokalem Rechner (z. B. MySQL Workbench 8.0)
- einen Benutzernamen und ein Passwort wurden festgelegt
- Neue Datenbank-Schema wurde in MySQL Workbench erstellt.
- Neue Tabelle wurde im neuen Schema erstellt.

Um die neu erstellte Tabelle mit den Daten einzupflegen, benötigen wir zuerst die Liste von allen Kraftfahrzeug (KFZ) Deutschlands. Die Daten für die Liste haben wir aus unserer Recherche unter <https://www.kennzeichenking.de/kfz-kennzeichen-liste> entnommen. Des Weiteren wurden anhand der in der URL abgebildete Liste folgendes durchgeführt:

- Erstellen eines Skripts, um die KFZ-Daten aus der Tabelle zu extrahieren. (Dieses Skript ist im Anhang unter Anhang A.5 - Skript zu finden.)
- Speichern der KFZ-Daten in Form einer JSON Datei.
- Importieren der JSON Datei in die MySQL Workbench, um die neu erstellte Tabelle mit Daten versorgen zu können.

Somit wurden unsere Datenbank mit der benötigten Daten eingepflegt und konnte dann in MySQL Workbench mit SQL-Abfragen getestet werden. Die nächste Vorgehensweise ist die Herstellung einer Verbindung zwischen Datenbank und Express-Back-End Anwendung. (Siehe Anhang A.6). An dieser Stelle wurden Folgendes durchgeführt:

- Installation Node.JS Module für MySQL Datenbank
- Verbindung mit localhost mit einem bestimmten Port wurde bestimmt
- Nachdem die Verbindung mit der Express-Back-End Anwendung erfolgreich hergestellt wurde, wurden SQL-Abfragen in Express-Back-End Anwendung abgerufen und getestet, ob die Inhalt der KFZ-Tabelle angezeigt werden kann. Ein Ausschnitt dazu ist im Anhang (A.7) zu finden.

Abschließend ist noch die Anbindung mit der React-Front-End Anwendung, um die Anwendung in tabellarisch darzustellen. Folglich mit etwas Cascading Style Sheets (CSS)-Anpassung sollte die Anwendung optisch besser und angenehm anzusehen sein.

3.2 Qualitätssicherung

Um die Qualität der abzuliefernden Webapplikation zu garantieren, werden User-Flow's definiert. Anhand diesen User-Flows werden Regressionstests durchgeführt. Eine Auflistung der User-Flows ist in der folgenden Tabelle einsehbar.

Nr.	User Flow
1	Laden der Seite
2	Einsehen der Übersicht mit allen Informationen
3	Ausgabe Kreis und Bundesland der eingegebenen Anfangsbuchstaben des Kraftfahrzeugkennzeichen
4	Ermittlung der Kennzeichen nach Bundesland
5	Ermittlung der Kennzeichen nach Kreis
6	Ermittlung der Kennzeichen nach Kreisstadt
7	Weiterleitung auf Wikipedia bei Klick auf den Wikipedia Button
7	Weiterleitung auf Google Maps bei Klick auf den Google-Maps Button
8	Export der Daten in XML
9	Export der Daten in CSV
10	Export der Daten in JSON
11	Import der Daten in XML
12	Import der Daten in CSV
13	Import der Daten in JSON
14	Hinzufügen von Datensätze

Tabelle 1: User Flow's

Zusätzlich zu der Tabelle ist unter A.1 ein Use-Case-Diagramm und unter A.2 die Beschreibung zu den einzelnen Anwendungsfällen hinterlegt.

3.3 Abweichungen & Anpassungen

Die vom Auftraggeber gestellten Forderungen wurden alle eingehalten. Somit gibt es keine besonderen Abweichungen und Anpassungen. Lediglich die gewünschte Zusatzfunktion des Auftraggebers wurde

zusätzlich implementiert. Nach einer gemeinsamen Besprechung der Projektbeteiligten wurde eine Funktion implementiert, welche den Benutzer erlaubt, Datensätze hinzuzufügen. Dadurch wird eine Anpassung der Datensätze vereinfacht. Weitere Informationen sind in der Benutzerdokumentation unter A.8 zu finden.

4 Projektergebnisse

4.1 Soll-/Ist-Vergleich

Die für das Projekt geplante Bearbeitungszeit, Personal und Sachmittel konnten eingehalten werden. Dabei ergaben sich kleinere Verschiebungen zwischen den Teil-Aufgaben, die im Abschnitt 3.1 Prozessschritte und Vorgehensweise aufgezählt werden. Der zeitliche Ablauf des Projektes ist aus der folgenden Tabelle ersichtlich.

Beispiel (verkürzt) Wie in Tabelle 2 zu erkennen ist, konnte die Zeitplanung bis auf wenige Ausnahmen bei der Dokumentation sowie das Erstellen der Datenbank eingehalten werden. Die benötigte Zeit für die Dokumentation, das Erstellen, sowie auch beim Einpflegen der Datenbank ist länger als vorher geschätzt wurde.

Phase	Beschreibung der Aufgaben	Soll-Stunden	Ist-Stunden	Differenz
Analyse	Anwendungsfallbeschreibung	1 h	1 h	
	Architekturauswahl	1 h	1 h	
Design	Datenbankdesign und Einrichtung	2 h	1 h	-1 h
	Erstellen des Use-Case- & Klassendiagramms	2 h	1 h	-1 h
Implementierungsphase	Erstellen der Datenbank	2 h	3 h	+1 h
	Erstellen der Back-Ends Anwendung	3 h	3 h	
	Erstellen der Front-End Anwendung	5 h	5 h	
	Testen der Anwendung	2 h	2 h	
Dokumentation	Schreiben der Projektdokumentation	5 h	6 h	+1 h
	Schreiben Benutzerdokumentation	1 h	1 h	
Rollout	Abnahme und Einführung	1 h	1 h	
Gesamt		25 h	25 h	0 h

Tabelle 2: Soll-/Ist-Vergleich

4.2 Lesson Learned

Im Projektverlauf konnten die Autoren, wertvolle Erfahrungen in der Projektplanung und -durchführung sammeln. Insbesondere die Bedeutung der kontinuierlichen Kommunikation untereinander, um für eine erfolgreiche Umsetzung des Projektes zu garantieren, wird deutlich. Zudem wurden neue Erkenntnisse zur Integration und Nutzung der angewandten Technologien gewonnen. Beispielsweise erwies sich das React-Framework als sehr nützlich, da man die Möglichkeit hat Komponenten einfach wiederzuverwenden. Zudem ist die Möglichkeit geboten, den Status einer Komponente in jedem einzelnen Moment zu

4 Projektergebnisse

überprüfen. Zusammenfassend lässt sich sagen, dass die Realisierung dieses Projekts einen deutlichen Mehrwert für alle beteiligten Autoren hat.

A Anhang

A.1 Use-Case-Diagramm

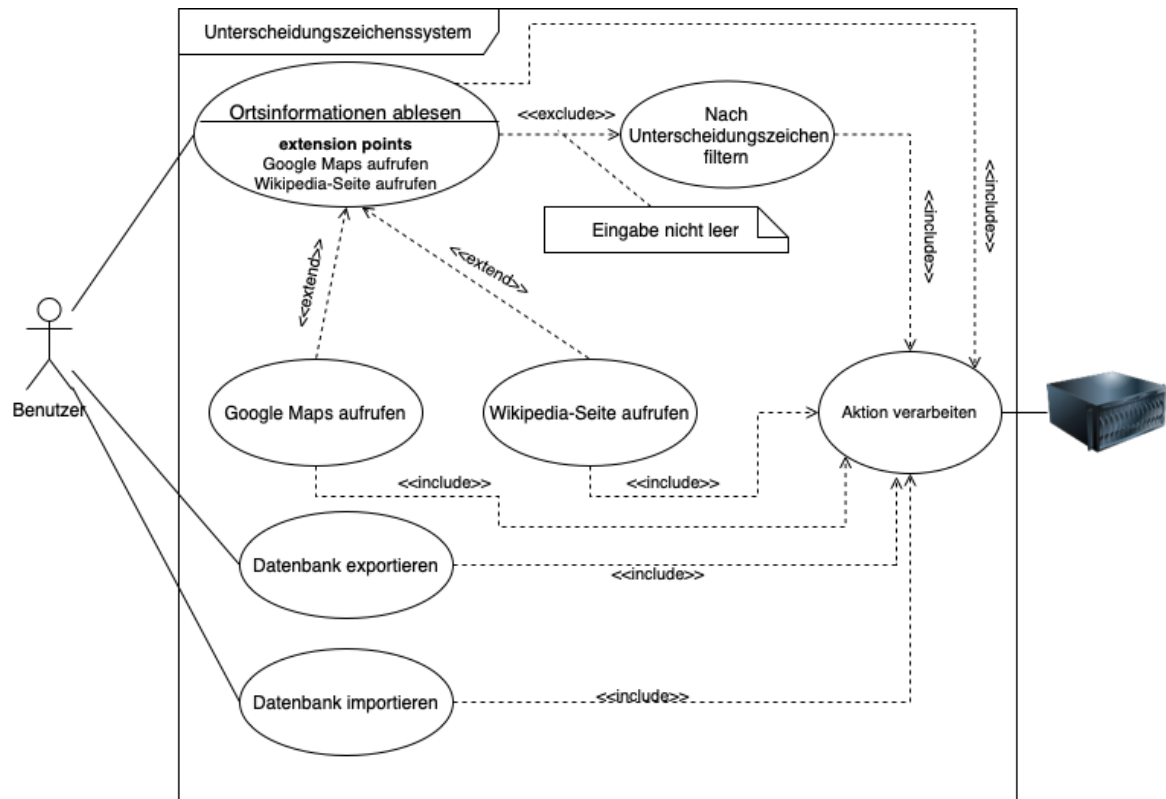


Abbildung 1: Use-Case Diagramm für die Anwendung KFZ-Kennzeichnung

A.2 Anwendungsfall-Beschreibung

Anwendungsfall-Beschreibungen

Beschreibung der Geschäftsprozesse

GP Eingabe: Unterscheidungszeichen

Ziel:	Durch die Eingabe des Unterscheidungszeichen wird der Geschäftsprozess "Ortsinformationen ablesen" eingeleitet
Vorbedingung	keine
Nachbedingung Erfolg:	Geschäftsprozesses "Ortsinformationen ablesen" wird eingeleitet, Eingabefeld wird geleert, Eingabe kann erneut getätigt werden
Akteur	Benutzer
Auslösendes Ereignis	Anzeigen der Ortsinformationen durch klicken des "Search-Buttons"
Beschreibung	<ol style="list-style-type: none">1. Unterscheidungszeichen in das Texteingabefeld eingeben2. Mausklick auf "Search-Button"3. Ortsinformationen werden angezeigt

GP Ortsinformationen ablesen

Ziel:	Anzeigen der Ortsinformationen
Vorbedingung	GP Eingabe: Unterscheidungszeichen
Nachbedingung Erfolg:	Anzeigen der Ortsinformationen, Weiterleitungslink werden angezeigt für: <ol style="list-style-type: none">1. Google Maps2. Wikipedia
Akteur	Benutzer
Auslösendes Ereignis	Eingabe: Unterscheidungszeichen
Beschreibung	<ol style="list-style-type: none">1. GP Eingabe: Unterscheidungszeichen2. Anzeigen der Ortsinformationen3. Anzeigen der Weiterleitungslinks für Google Maps und Wikipedia

GP Datenbank importieren

Ziel:	Datenbank wird mit der zu importierenden Datenbank ersetzt
Vorbedingung	Zu importierende Datei wurde ausgewählt, Datei muss das Format XML, CSV oder JSON haben
Nachbedingung Erfolg:	Alte Datenbank wird gelöscht, Neue Datenbank wird importiert, Meldung das Importieren erfolgreich oder fehlgeschlagen ist
Akteur	Benutzer
Auslösendes Ereignis	Importierung durch Auswählen der Datei und Mausklick auf den "Import-Button"
Beschreibung	<ol style="list-style-type: none">1. Datei auswählen2. Mausklick auf "Import-Button"3. Alte Datenbank wird mit der neuen Datenbank ersetzt.4. Meldung erfolgreich/fehlgeschlagen

GP Datenbank exportieren

Ziel:	Datenbank wird als XML, CSV oder JSON Datei exportiert
Vorbedingung	Dateiformat wurde ausgewählt
Nachbedingung Erfolg:	Datei wird heruntergeladen
Akteur	Benutzer
Auslösendes Ereignis	Exportierung durch Mausklick auf den "Export-Button"
Beschreibung	<ol style="list-style-type: none">1. Auswahl des gewünschten Formates2. Mausklick auf "Export-Button"3. Datei wird heruntergeladen

A.3 GANTT-Diagramm



Abbildung 2: GANTT-Diagramm zur Visualisierung der Schritte zum Projektziel

A.4 Einpflegen der Datenbank-Tabelle

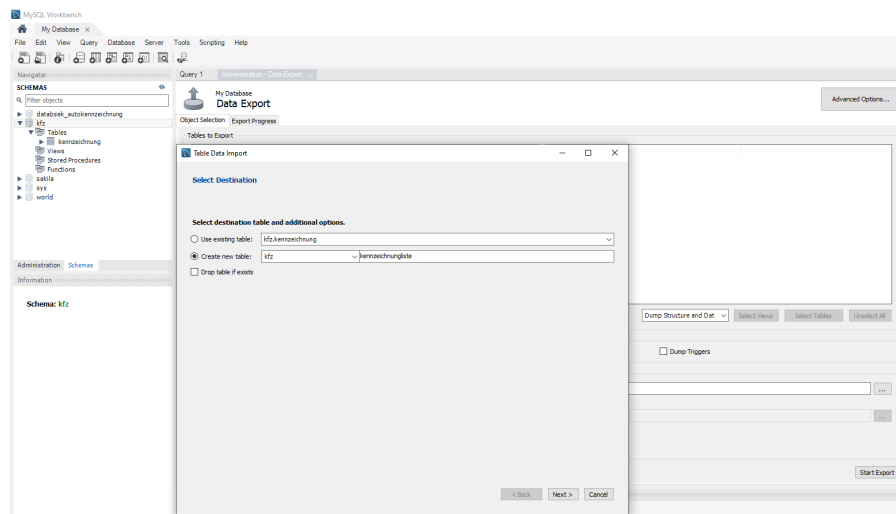


Abbildung 3: Das Laden der JSON Datei in die neue Tabelle

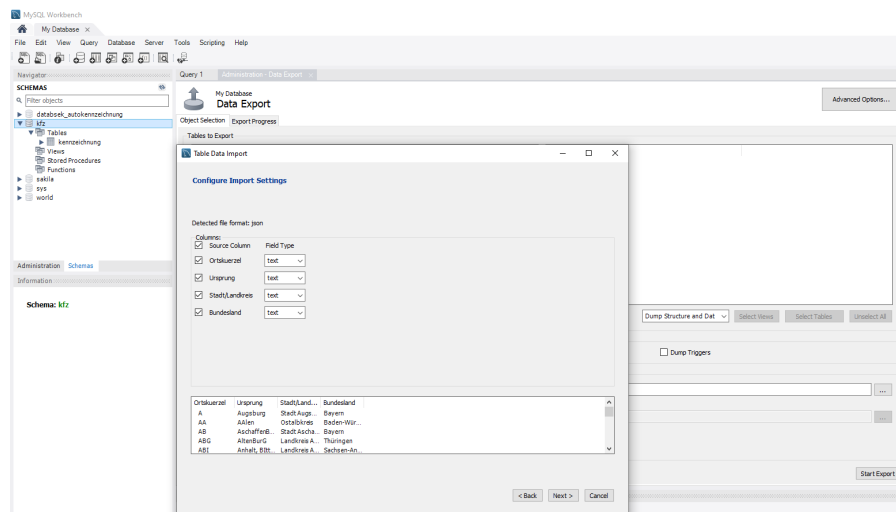


Abbildung 4: Konfigurierung der Tabelle

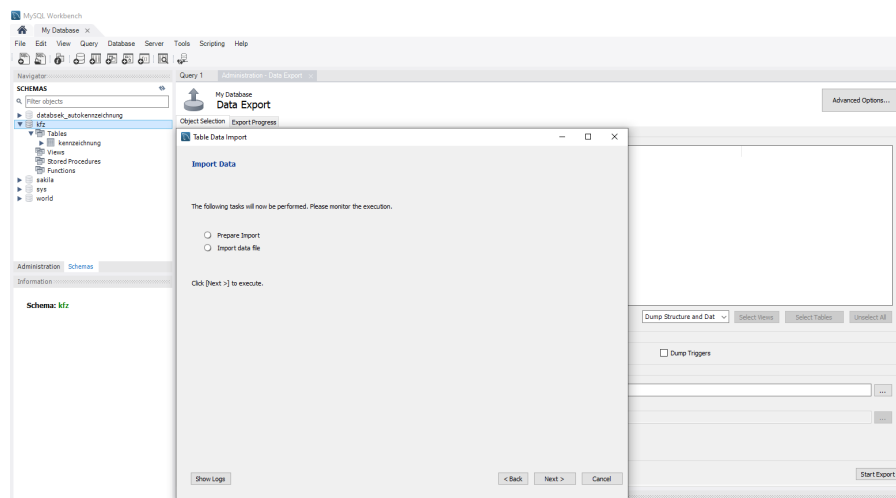


Abbildung 5: Beginn des Importierens

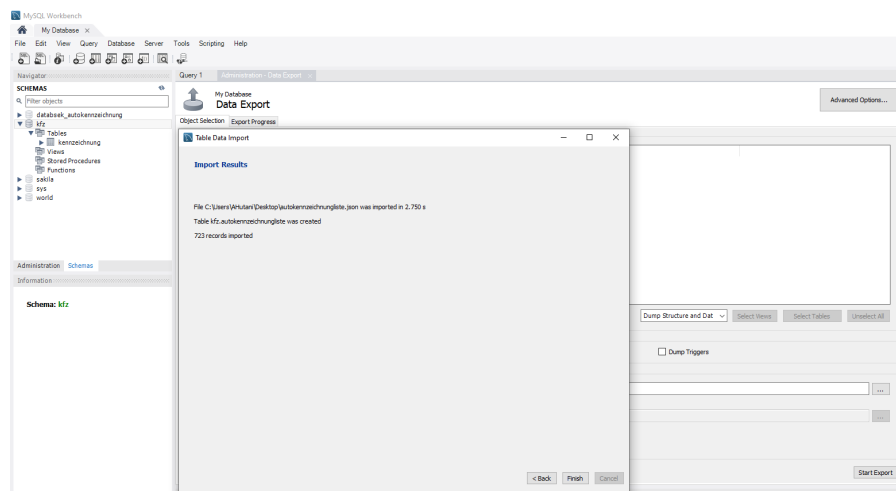


Abbildung 6: Beginn des Importierens

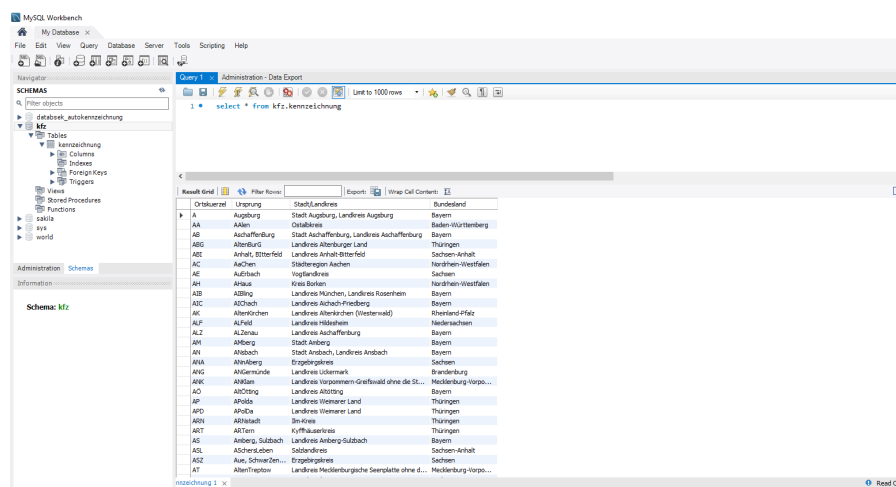


Abbildung 7: Ansicht der Tabelle KFZ Kennzeichnung

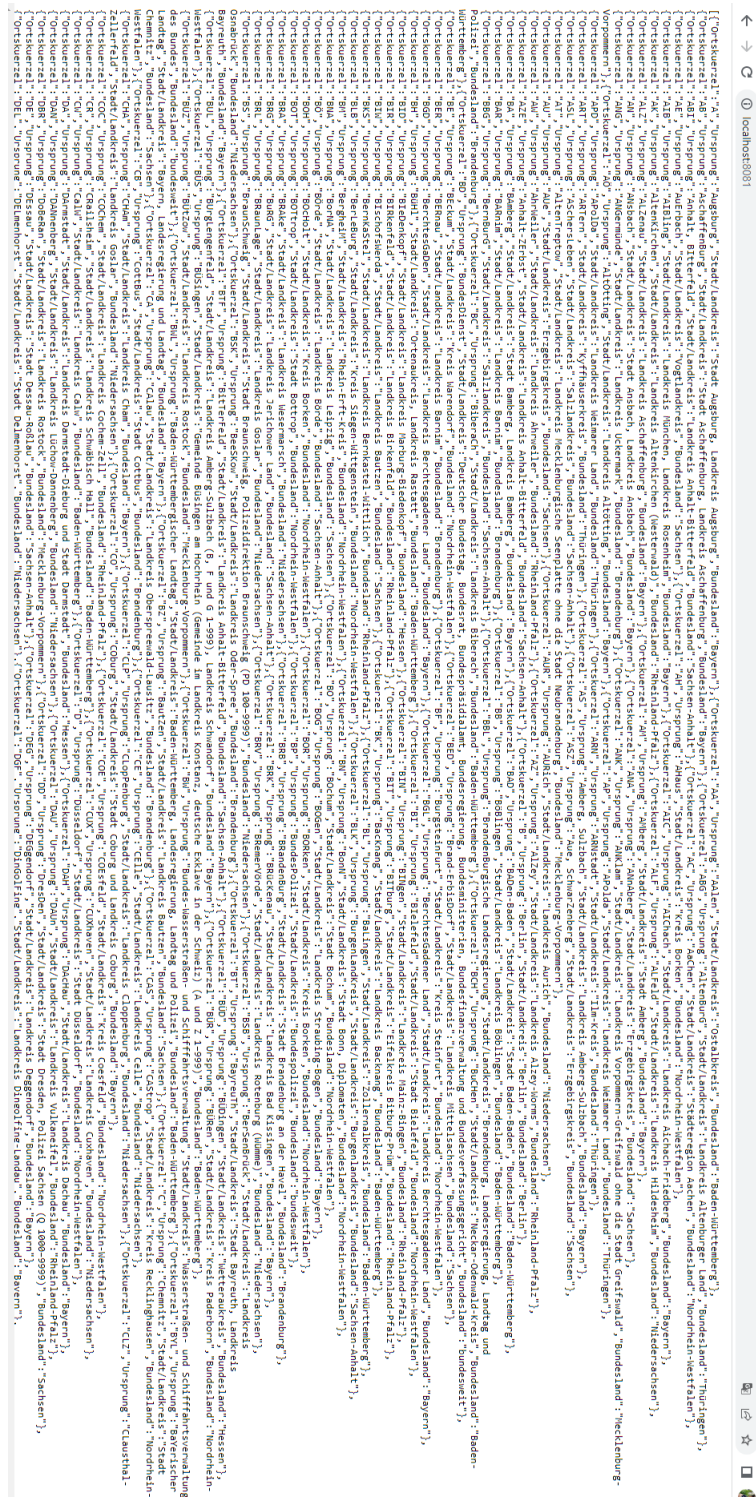


Abbildung 8: Ansicht der Datenbank-Inhalt in Back-End-Anwendung

A.5 Skript

```
1  let rows = document.querySelectorAll("table tbody tr");
2  let output = [];
3  rows.forEach(element => {
4      let cells = element.querySelectorAll("td");
5      let entity = {
6          Ortskuerzel: cells[0].textContent,
7          Ursprung: cells[1].textContent,
8          "Region": cells[2].textContent,
9          Bundesland: cells[3].textContent
10     }
11     output.push(entity);
12 });
13 console.log(output)
```

Abbildung 9: Skript zum Extrahieren der KFZ-Kennzeichen Daten

A.6 Datenbank Anbindung mit Express-Back-End

Folgendes ist der Ausschnitt der Verbindung zwischen MySQL und ExpressJS

```
1 var mysql = require('mysql2');
2
3 var connection = mysql.createConnection({
4   host: 'localhost',
5   user: 'root',
6   password: 'root',
7   database: 'kfz'
8 });
9
10 module.exports = connection;
```

Listing 1: Javascript Code für die Herstellung der Verbindung mit Datenbank

A.7 Abrufen der SQL-Abfrage in Back-End Anwendung

Folgendes ist der Ausschnitt vom Abrufen der SQL Abfragen

```
1 const express = require("express");
2 const bodyParser = require("body-parser");
3 const app = express();
4 const connection = require('./database');
5 const cors = require('cors');
6 const fs = require("fs");
7 const path = require("path");
8 const multer = require('multer')
9 const convert = require('xml-js');
10 const { parse } = require('json2csv');
11 const csv = require('csvtojson');
12 const upload = multer({
13   dest: 'uploads/'
14 })
15 const parser = require('xml2json')
16
17 var corsOptions = {
18   allowedHeaders: 'Content-Type, requested-type'
19 }
20
21 app.use(cors(corsOptions));
22 app.use(bodyParser.json());
23
24 app.get('/', function (req, res) {
25   let sql = "SELECT * FROM kennzeichnung ORDER BY Ortskuerzel ASC";
26   connection.query(sql, function (err, results, fields) {
27     if (err) throw err;
28     res.send(results);
```

```

29     });
30 });
31
32 app.post('/importFromFile', upload.single('uploadedFile'), async function (req, res, next) {
33     const absolutePath = path.join(__dirname, req.file.path);
34     let jsonObject;
35
36     if(req. file .mimetype == "application/json"){
37         const jsonString = fs.readFileSync(absolutePath, "utf-8");
38         jsonObject = JSON.parse(jsonString);
39     } else if (req. file .mimetype == "text/csv" || req.file.mimetype == "application/vnd.ms-excel") {
40         await csv().fromFile(absolutePath).then((jsonObj) => {
41             jsonObject = jsonObj;
42         })
43     } else if (req. file .mimetype == "text/xml"){
44         const xmlString = fs.readFileSync(absolutePath, "utf-8");
45         jsonObject = parser.toJson(xmlString)
46         jsonObject = JSON.parse(jsonObject).elements
47         console.log(jsonObject)
48     }
49     await importLicensePlates(jsonObject, res);
50 })
51
52 app.post('/addLicensePlate/:Ortskuerzel/:Ursprung/:Landkreis/:Bundesland', function (req, res) {
53     const values = [req.params.Ortskuerzel, req.params.Ursprung, req.params.Landkreis, req.params.Bundesland]
54     let query = 'INSERT INTO kennzeichnung (ortskuerzel, ursprung, landkreis, bundesland) VALUES (?, ?, ?, ?)';
55     connection.query(query, values, function(err, result) {
56         if(err) {
57             console.log(err)
58         } else {
59             res.send("SUCCESS")
60         }
61     })
62 })
63
64 async function importLicensePlates(requestBody, res) {
65     let values = []
66     Object.keys(requestBody).forEach((val, i) => {
67         values.push([requestBody[val].Ortskuerzel, requestBody[val].Ursprung, requestBody[val]["Region"],
68             requestBody[val].Bundesland]);
69     })
70     connection.query('TRUNCATE TABLE kennzeichnung', function (err, result) {
71         if (err) {
72             res.send('Error Truncate table');
73         } else {
74             connection.query('INSERT INTO kennzeichnung (ortskuerzel, ursprung, landkreis, bundesland)
75                 VALUES ?', [values], function (err, result) {
76                 if (err) {
77                     res.send(err);
78                 } else {

```

```

77         res.send('Success');
78     }
79 })
80 }
81 })
82 }
83
84 async function exportLicensePlates(req, res) {
85     let query = 'SELECT Ortskuerzel as Ortskuerzel, Ursprung as Ursprung, Landkreis AS 'Region', Bundesland AS
86         Bundesland FROM kennzeichnung ORDER BY Ortskuerzel ASC'
87     connection.query(query, function (err, result) {
88         if (err) {
89             console.log(err)
90             res.send('Error while exporting JSON file from the database');
91         } else {
92             if (req.get("Requested-Type") === "application/json") {
93                 res.writeHead(200, { 'Content-Type': 'application/json' })
94                 res.write(JSON.stringify(result))
95                 res.end()
96             } else if (req.get("Requested-Type") === "text/xml") {
97                 let formattedResult = []
98                 formattedResult["elements"] = {}
99                 result.forEach((value, index) => {
100                     formattedResult["elements"]["elem"+index] = value
101                 });
102                 res.send(convert.json2xml(formattedResult, {
103                     compact: true,
104                     ignoreComment: true,
105                     spaces: 4
106                 }))
107             } else if (req.get("Requested-Type") === "text/csv") {
108                 res.send(parse(result))
109             }
110         }
111     })
112 }
113 app.post("/import", function (req, res) {
114     importLicensePlates(req.body, res)
115 })
116
117 app.get("/export", async function (req, res) {
118     await exportLicensePlates(req, res)
119 })
120
121 app.get('/ursprung/:ursprungName', function (req, res) {
122     let ursprungName = decodeURI(req.params.ursprungName)
123     var sql = "SELECT * FROM kennzeichnung WHERE ursprung LIKE concat('%', ?, '%')";
124     connection.query(sql, ursprungName, function (err, results, fields) {
125         if (err) throw err;

```

```

126     res.send(results);
127   });
128 });
129
130 app.get('/deletedb', function (req, res) {
131   console.log("Deleting DB...")
132   connection.query('TRUNCATE TABLE kennzeichnung', function (err, result) {
133     if (err) {
134       res.send('Error Truncate table');
135     } else {
136       res.send("SUCCESS")
137     }
138   })
139 });
140
141 app.get('/ortskuerzel/:kuerzel', function (req, res) {
142   let kuerzel = req.params.kuerzel
143   var sql = "SELECT * FROM kennzeichnung WHERE ortskuerzel LIKE concat('%', ?, '%') ORDER BY
144     Ortskuerzel ASC";
145   connection.query(sql, kuerzel, function (err, results, fields) {
146     if (err) throw err;
147     res.send(results);
148   });
149 });
150
151 app.get('/landkreis/:landkreis', function (req, res) {
152   let landkreis = decodeURI(req.params.landkreis)
153   var sql = "SELECT * FROM kennzeichnung WHERE landkreis LIKE concat('%', ?, '%') ORDER BY Ortskuerzel
154     ASC";
155   connection.query(sql, landkreis, function (err, results, fields) {
156     if (err) throw err;
157     res.send(results);
158   });
159 });
160
161 app.get('/landkreis', function (req, res) {
162   var sql = "SELECT * FROM kennzeichnung ORDER BY landkreis ASC ";
163   connection.query(sql, function (err, results, fields) {
164     if (err) throw err;
165     res.send(results);
166   });
167 });
168
169 app.get('/bundesland/:bundesland', function (req, res) {
170   let bundesland = decodeURI(req.params.bundesland)
171   var sql = "SELECT * FROM kennzeichnung WHERE bundesland LIKE concat('%', ?, '%') ORDER BY
172     Ortskuerzel ASC";
173   connection.query(sql, bundesland, function (err, results, fields) {
174     if (err) throw err;
175     console.log(results)

```


```
173
174     res.send(results);
175   });
176 });
177
178 var server = app.listen(8081, function () {
179   var host = server.address().address
180   var port = server.address().port
181
182   console.log("Example app listening at http://%s:%s", host, port)
183 })
```

Listing 2: Javascript Code für bestimmten Datensatz aus der Tabelle abzufragen

A.8 Benutzerdokumentation

Ausschnitte aus der Benutzerdokumentation:



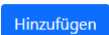




Ansicht der Anwendung:



Ortskürzel	Ursprung	Stadt/Land	Bundesland	Link
A	Augsburg		Bayern	Mapswiki
AA	Aalen		Baden-Württemberg	Mapswiki
AB	Aschaffenburg		Bayern	Mapswiki
ABG	Altenburg		Thüringen	Mapswiki
ABI	Anhalt, Bitterfeld		Sachsen-Anhalt	Mapswiki
AC	Aachen		Nordrhein-Westfalen	Mapswiki
AE	Aufbach		Sachsen	Mapswiki
AH	Ahaus		Nordrhein-Westfalen	Mapswiki
AIB	Aibling		Bayern	Mapswiki
AIC	Aichach		Bayern	Mapswiki
AK	Altenkirchen		Rheinland-Pfalz	Mapswiki
ALF	Alfeld		Niedersachsen	Mapswiki
ALZ	Alzenau		Bayern	Mapswiki
AM	Amberg		Bayern	Mapswiki
AN	Ansbach		Bayern	Mapswiki
ANA	Annaberg		Sachsen	Mapswiki
ANG	ANGermünde		Brandenburg	Mapswiki
ANK	ANKlam		Mecklenburg-Vorpommern	Mapswiki
AÖ	Altötting		Bayern	Mapswiki

Abbildung 10: Die Ansicht der Anwendung des KFZ-Verwaltungssystems

Funktion der vorhandenen Buttons:

Symbol	Bedeutung	Funktion
	Dropdown Export	Die Dropdown enthält die Auswahl zur verschiedenen Datenformate, z. B.: CSV, JSON, XML.
	Import	Hier können verschiedene Datenformate wie z.B. : CSV, JSON, XML aus lokalem PC importiert werden.
	Hinzufügen	Benutzer kann neuen Datensatz in die Tabelle einfügen.
	Such-Button	Sie gibt dem Benutzer den gesuchten Datensatz zurück.
	Reset-Button	Sie setzt alle gefüllten bzw. vorgewählten Felder, wie Suchfeld, Dropdown Spalte, wieder zurück.
	Dropdown-Filter	Der Benutzer kann das Suchergebnis in die engere Auswahl treffen, ob sie nach Ortskuerzel, Ursprung, oder Bundesland gefiltert werden soll.
	WikiLink	Ein externer Link, der den Benutzer zum Wikipedia leitet.

Anleitung:

Der folgende Link führt Sie zum Quellcode in GitHub:

<https://github.com/Piotr-Dziekonski/autokennzeichen-tool.git>

<https://github.com/Piotr-Dziekonski/autokennzeichen-tool-backend.git>

In den einzelnen Readme-Dateien der Repositories finden Sie die genaue Anleitung um das Projekt erfolgreich zu starten.