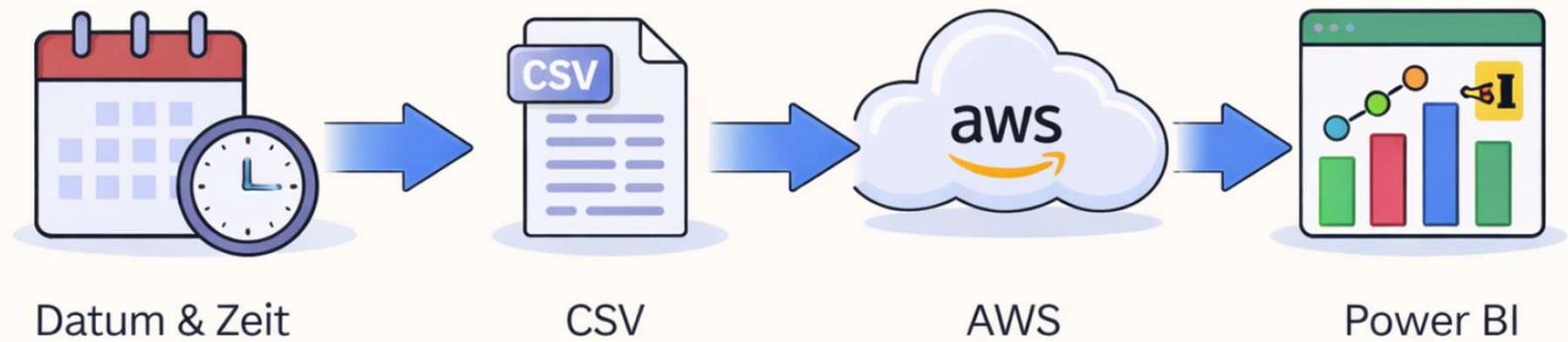
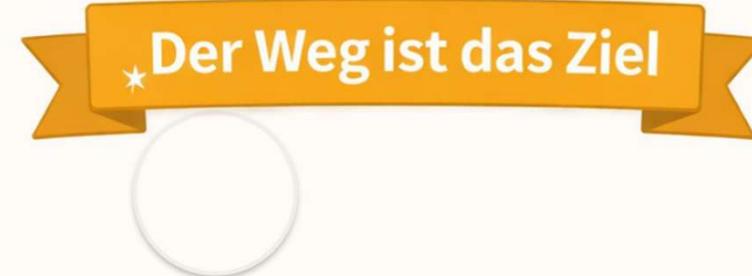


- Datum/zeit
- ETL/AWS
- Power BI



Von der manuellen Dateneingabe
zur strukturierten Analyse in Power BI

- Projekt zur Analyse meiner Lernzeiten
- Ziel: Lernzeiten sammeln, übertragen datenbasiert auswerten
- **Technologien:**
 - Excel, python
 - CSV
 - AWS
 - Power BI

„Der Weg ist das Ziel“

Das Ziel war nicht ein Dashboard-Ergebnis, sondern das Verständnis des gesamten Datenprozesses

Von der manuellen Zeiteingabe zur strukturierten Analyse in Power BI

Technologien:

- ⌚ Datum & Zeit in Excel, später Streamlit/Python-Eingabemaske
- CSV
- AWS
- Power BI

Das Motto meines Projekts lautet: *Der Weg ist das Ziel.*

Mir ging es nicht nur um das End-Dashboard, sondern darum, eine komplette Datenpipeline (ETL) aufzubauen – von der ersten Eingabe von Datum und Zeit bis zur professionellen Auswertung in Power BI

Ich wollte verstehen, wie Daten in der Praxis ihren Weg nehmen – von der Erfassung über die Cloud bis zur Analyse. Deshalb habe ich bewusst eine komplette End-to-End-Pipeline aufgebaut

Ausgangssituation

- Mehrere Excel-Dateien
- Unterschiedliche Formate
- Keine klare Auswertung

Am Anfang habe ich meine Lernzeiten in mehreren Excel-Dateien erfasst. Die Formate waren unterschiedlich und Auswertungen waren kaum möglich. Ich konnte nicht sehen, wann oder wie effizient ich gelernt habe

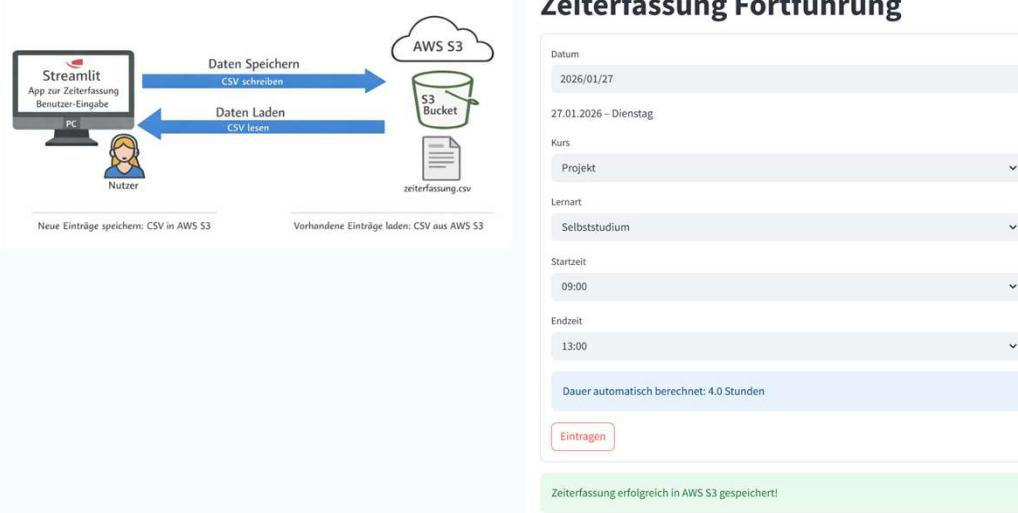
	A	B	C	D	E	F	G	H	I
	Datum	Wochentag	Kursname	Lernart	Thema / Inhalt	Startze	Endze	Dauer (h)	Lernmodus / Quelle
1	31.03.2025	5	2	summe					
2	01.04.2025	6							
3	02.04.2025	7							
4	03.04.2025	6	2	8					
5	04.04.2025			0					
6	05.04.2025			0					
7	06.04.2025			0					
				Wochenstunden:	28				
8									
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									
20									
21									
22									
23									
24									
25									
26									
27									
28									
29									
30									
31									
32									
33									
34									
35									
36									
37									
38									
39									
40									
41									
42									
43									
44									
45									
46									
47									
48									
49									
50									

	A	B	C	D	E	F	G	H	I	
	Datum	Wochentag	Kursname	Lernart	Thema / Inhalt	Startze	Endze	Dauer (h)	Lernmodus / Quelle	
1	01.10.2025	Mittwoch	EXCEL	Selbststudium	Hausaufgaben	10:00	12:00	2,00	DSI	
2	02.10.2025	Donnerstag	EXCEL	Live-Call-Vorlesung	Live-Teilnehmen	13:15	14:50	1,58	DSI	
3	03.10.2025	Freitag	EDA	Selbststudium	Hausaufgaben	09:00	12:00	3,00	DSI	
4	04.10.2025	Samstag	EDA	Theorie (Video)	Hören/Sehen	09:00	14:00	5,00	DSI	
5	05.10.2025	Sonntag	EDA	Theorie (Video)	Hören/Sehen	09:00	13:00	4,00	DSI	
6	06.10.2025	Montag	EXCEL	Live-Call-Hausaufgaben	Live-Teilnehmen	10:00	11:00	1,00	DSI	
7	07.10.2025	Dienstag	ScrumPO	Selbststudium	Prüfungsvorbereitung	14:00	19:00	5,00	Udemy	
8	08.10.2025	Mittwoch	EDA	Live-Call-Vorlesung	Live-Teilnehmen	10:15	11:45	1,50	DSI	
9	09.10.2025	Donnerstag	PROJEKT	Selbststudium	Zeitfassung	08:30	10:45	2,25	Excel	
10	10.10.2025	Freitag	PROJEKT	Selbststudium	Zeitfassung	16:45	18:15	1,50	Udemy	
11	11.10.2025	Samstag	EDA	Selbststudium	Prüfungsvorbereitung	09:30	11:00	1,50	Udemy	
12	12.10.2025	Sonntag	ScrumPO	Selbststudium	Zeitfassung	11:00	13:30	2,50	Python, Streamlit, GPT	
13	13.10.2025	Montag	ETL	Selbststudium	Prüfungsvorbereitung	16:30	18:00	1,50	Udemy	
14	14.10.2025	Dienstag	AWS	Selbststudium	PowerBI	Hören/Sehen	08:40	09:00	0,33	DSI
15	15.10.2025	Mittwoch	ScrumPO	Selbststudium	Prüfungsvorbereitung	13:00	14:30	1,50	Udemy	
16	16.10.2025	Freitag	PROJEKT	Selbststudium	Prüfungsvorbereitung	16:45	18:00	1,25	Udemy	
17	17.10.2025	Samstag	EDA	Live-Call-Hausaufgaben	Live-Teilnehmen	09:15	10:35	1,00	DSI	
18	18.10.2025	Sonntag	ScrumPO	Selbststudium	Prüfungsvorbereitung	10:45	12:15	1,50	DSI	
19	19.10.2025	Montag	ETL	Live-Call-Vorlesung	Live-Teilnehmen	14:00	15:35	1,25	Udemy	
20	20.10.2025	Dienstag	AWS	Selbststudium	Prüfungsvorbereitung	10:00	11:20	1,33	Udemy	
21	21.10.2025	Mittwoch	PowerBI	Selbststudium	Prüfungsvorbereitung	10:00	11:00	1,00	Udemy	
22	22.10.2025	Freitag	PROJEKT	Selbststudium	Prüfungsvorbereitung	11:00	16:35	5,25	DSI	
23	23.10.2025	Samstag	EDA	Live-Call-Hausaufgaben	Live-Teilnehmen	09:00	14:30	5,50	DSI	
24	24.10.2025	Sonntag	ScrumPO	Selbststudium	Prüfungsvorbereitung	10:00	13:30	3,50	Udemy	
25	25.10.2025	Montag	ETL	Theorie (Video)	Hören/Sehen	16:30	18:15	1,75	DSI	
26	26.10.2025	Dienstag	ScrumPO	ETL	Theorie (Video)	Hören/Sehen	09:00	14:30	5,50	DSI
27	27.10.2025	Mittwoch	ScrumPO	Selbststudium	Prüfungsvorbereitung	17:00	18:00	1,00	Udemy	
28	28.10.2025	Freitag	EDA	Live-Call-Vorlesung	Live-Teilnehmen	10:15	12:00	1,75	DSI	
29	29.10.2025	Samstag	ScrumPO	Selbststudium	Prüfungsvorbereitung	15:00	16:00	1,00	Udemy	
30	30.10.2025	Sonntag	ScrumPO	Selbststudium	Prüfungsvorbereitung	10:00	11:20	1,33	Udemy	
31	31.10.2025	Montag	ETL	Selbststudium	Prüfungsvorbereitung	10:00	11:00	1,00	Udemy	
32	32.10.2025	Dienstag	ScrumPO	ETL	Theorie (Video)	Hören/Sehen	11:00	16:35	5,25	DSI
33	33.10.2025	Mittwoch	ScrumPO	Selbststudium	Prüfungsvorbereitung	10:00	11:00	1,00	Udemy	
34	34.10.2025	Freitag	ScrumPO	Selbststudium	Prüfungsvorbereitung	10:00	13:30	3,50	Udemy	
35	35.10.2025	Samstag	ETL	Live-Call-Vorlesung	Live-Teilnehmen	10:15	12:00	1,75	DSI	
36	36.10.2025	Sonntag	ETL	Theorie (Video)	Hören/Sehen	13:00	14:00	1,00	DSI	
37	37.10.2025	Montag	ETL	Theorie (Video)	Hören/Sehen	15:00	16:30	1,50	DSI	
38	38.10.2025	Dienstag	ScrumPO	Selbststudium	Prüfungsvorbereitung	09:00	11:00	2,00	Stepik	
39	39.10.2025	Mittwoch	ETL	Theorie (Video)	Hören/Sehen	11:30	13:30	2,00	DSI	
40	40.10.2025	Freitag	ScrumPO	Theorie (Video)	Hören/Sehen	14:00	17:00	3,00	Udemy	
41	41.10.2025	Samstag	ScrumPO	Selbststudium	Prüfung	10:00	11:30	1,50	Scrum	
42	42.10.2025	Sonntag	PowerBI	Theorie (Video)	Hören/Sehen	09:30	14:00	4,50	DSI	
43	43.10.2025	Montag	PowerBI	Theorie (Video)	Hören/Sehen	15:00	17:00	2,00	DSI	
44	44.10.2025	Dienstag	PowerBI	Theorie (Video)	Hören/Sehen	09:00	12:30	3,50	DSI	
45	45.10.2025	Mittwoch	PowerBI	Live-Call-Vorlesung	Live-Teilnehmen	13:15	15:00	1,75	DSI	
46	46.10.2025	Freitag	PowerBI	Theorie (Video)	Hören/Sehen	09:00	10:00	1,00	Stepik	
47	47.10.2025	Samstag	ETL	Live-Call-Vorlesung	Live-Teilnehmen	10:15	11:45	1,50	DSI	
48	48.10.2025	Sonntag	PowerBI	Theorie (Video)	Hören/Sehen	12:00	14:00	2,00	DSI	
49	49.10.2025	Montag	PowerBI	Selbststudium	Hausaufgaben	11:30	14:00	2,50	DSI	
50	50.10.2025	Dienstag	PowerBI	Selbststudium	Hausaufgaben	16:30	21:00	4,50	DSI	

- Benutzeroberfläche
- Automatische Stundenberechnung
- Einheitliches Format

Streamlit

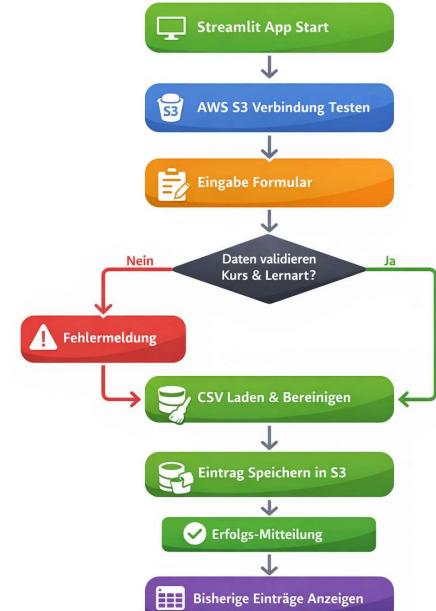
Die CSV-Datei wird nicht lokal verwaltet. Streamlit dient ausschließlich als Frontend zur Dateneingabe. Hier trage ich Datum, Start- und Endzeit ein, und die Dauer wird automatisch berechnet. Die Persistenz erfolgt zentral in AWS S3, wodurch die Anwendung zustandslos bleibt und sich problemlos in eine Cloud-Pipeline mit Glue, Athena und Power BI integrieren lässt



Bisherige Einträge

	Datum	Wochentag	Kursname	Lernart	Startzeit	Endzeit	Dauer (h)
354	27.01.2026	Dienstag	Projekt	Selbststudium	09:00	13:00	4
353	27.01.2026	Dienstag	Karrierecoaching	Selbststudium	22:00	23:45	1.75
352	26.01.2026	Montag	Projekt	Selbststudium	17:00	18:00	1
351	26.01.2026	Montag	Karrierecoaching	Selbststudium	18:00	20:00	2
350	26.01.2026	Montag	Karrierecoaching	Selbststudium	18:00	20:00	2

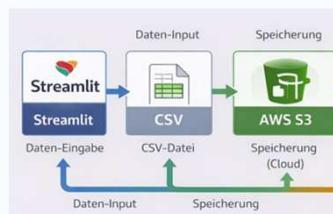
Zeiterfassungs-App mit Streamlit & AWS S3



- Zentrale Datenablage
- Trennung von Speicherung & Analyse
- Grundlage für BI

AWS S3

Hier sieht man den technischen Kern meiner Datenpipeline. Die Streamlit-App erzeugt aus den erfassten Zeiten eine CSV-Datei und speichert diese automatisiert in einem AWS-S3-Bucket. S3 übernimmt dabei die Rolle eines zentralen Datenspeichers und bildet die Grundlage für die spätere Analyse in Power BI



```
# =====
# AWS S3 KONFIGURATION
#
# =====
import boto3 # AWS SDK für Python, Zugriff auf S3
S3_BUCKET = "zeiterfassung-bucket" # Name des S3-Buckets
S3_KEY = "zeiterfassung/zeiterfassung.csv"
# Pfad/Key innerhalb des Buckets

# boto3-Client erstellen
s3 = boto3.client("s3", region_name="eu-central-1")
```

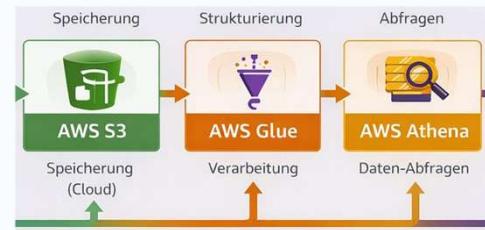
Warum nicht direkt Power BI?

S3 sorgt dafür, dass die Datenspeicherung von der Visualisierung getrennt ist – das ist robuster und näher an realen Datenpipelines.

Ein Screenshot der AWS S3 Konsole zeigt die Liste der Objekte im Bucket 'zeiterfassung-bucket'. Es sind 1 Objekt vorhanden: 'zeiterfassung.csv' (Typ: csv, Größe: 15.8 KB, Speicherklasse: Standard). Die Liste ist nach Name sortiert. Oben rechts befindet sich ein Button 'S3-URI kopieren'.

- Glue: Schema-Erkennung
- Athena: SQL-Abfragen auf CSV

AWS Glue & Athena (1)



The screenshots illustrate the AWS Glue interface:

- Raupen:** Shows the crawler configuration for "zeiterfassung_db". It lists crawlers like "zeiterfassung-CSV-Crawler" and "zeiterfassung_b-CSV-Crawler", along with their status (e.g., "Bereit", "Fehler", "Ausgeführt", "Fehlerhaft").
- Datenkatalog:** Shows the "zeiterfassung_db" database in the Data Catalog. It displays the table "zeiterfassung" with its schema ("Zeiterfassung aus Streamlit (CSV auf S3)"), and a list of tables including "zeiterfassung" and "zeiterfassung_b".
- Datenbanken:** Shows the "zeiterfassung_db" database in the Data Catalog, listing the tables "zeiterfassung" and "zeiterfassung_b".

AWS Glue erkennt automatisch die Struktur der Daten. Athena erlaubt mir dann, mit SQL direkt auf den CSV-Dateien zu arbeiten – ohne eine klassische Datenbank zu betreiben

Warum AWS Glue?

AWS Glue erstellt aus den CSV-Dateien **automatisch ein Tabellenschema**

Spalten, Datentypen und Trennzeichen werden erkannt

Die CSV-Dateien werden dadurch wie eine Datenbanktabelle nutzbar

Glue fungiert als **Metadaten- und Schema-Schicht** zwischen S3 und Analyse

The screenshot shows the AWS Athena interface for the table "s3_Zeiterfassung".

- Tabellenübersicht:** Shows the table "s3_Zeiterfassung" with the status "Datenqualität - neu".
- Tabellendetails:** Displays the table's name ("s3_Zeiterfassung"), database ("zeiterfassung_db"), and location ("s3://zeiterfassung-bucket/zeiterfassung/").
- Einstellung:** Shows the table is in CSV format.
- Erweiterte Eigenschaften:** Shows the table was last updated on December 18, 2025, at 14:17:58.
- Schema:** Shows the table has 7 columns with the following schema:

#	Spaltenname	Datentyp
1	Datum	Zeichenkette
2	Wochentag	Zeichenkette
3	Kursname	Zeichenkette
4	Lernart	Zeichenkette
5	Startzeit	Zeichenkette
6	Endzeit	Zeichenkette
7	Dauer (h)	doppelt

- Glue: Schema-Erkennung
- Athena: SQL-Abfragen auf CSV

AWS Glue & Athena (2)

AWS Glue erkennt automatisch die Struktur der Daten. Athena erlaubt mir dann, mit SQL direkt auf den CSV-Dateien zu arbeiten – ohne eine klassische Datenbank zu betreiben

Warum Amazon Athena?

Athena erlaubt **SQL-Abfragen direkt auf S3-Daten**. Keine eigene Datenbank nötig
Ideal für **Analyse, Aggregationen und Reporting**. Power BI kann **direkt über den Athena-Connector** angebunden werden

Athena ist die **Analyse-Engine** für meine Zeiterfassungsdaten

Warum nicht DynamoDB?

DynamoDB ist für **Echtzeit-Anwendungen** optimiert. Komplexe Auswertungen (Gruppierungen, Zeitvergleiche, SQL-Analysen) sind dort schwieriger
Für **Reporting & BI** ist Athena **einfacher, transparenter und günstiger**
Athena passt besser zu Reporting und Analyse, DynamoDB eher zu Live-Apps



Glue sorgt dafür, dass meine CSV-Dateien aus S3 wie richtige Tabellen behandelt werden. Athena nutzt diese Struktur, um SQL-Abfragen auszuführen, die ich direkt in Power BI verwenden kann. So entsteht ein sauberer und professioneller Datenworkflow

Das Bild zeigt den Screenshot des Amazon Athena Abfrage-Editors. Oben ist das Navigationsmenü mit 'Amazon Athena > Abfrage-Editor' und verschiedene Registerkarten wie 'Editor', 'Aktuelle Abfragen', 'Gespeicherte Abfragen' und 'Einstellungen' zu sehen. Ein Hinweis über Typeahead-Vorschläge ist angezeigt. In der linken Spalte befindet sich das 'Daten'-Panel mit 'Datenquelle: AwsDataCatalog', 'Katalog: Keine' und 'Datenbank: zeiterfassung_db'. Die rechte Spalte zeigt die 'Abfrage 1' mit dem folgenden SQL-Code:

```
1 SELECT *
2 FROM "zeiterfassung_db"."s3_zeiterfassung"
3 ORDER BY date_parse(`datum`, '%d.%m.%Y') DESC
4 LIMIT 10;
```

Unten rechts befindet sich das 'Abfrageergebnisse'-Panel mit dem Titel 'Ergebnisse (10)'. Es zeigt 10 Zeilen mit den folgenden Werten:

#	datum	wochentag	kursname	lernart	startzeit	endzeit	dauer (h)
1	21.01.2026	Mittwoch	Karrierecoaching	Selbststudium	10:30	14:00	3.5
2	20.01.2026	Dienstag	Karrierecoaching	Selbststudium	18:00	20:00	2.0
3	20.01.2026	Dienstag	Karrierecoaching	Selbststudium	10:30	14:30	4.0

- Streamlit → CSV
- AWS S3 → Data Lake
- Glue & Athena → Struktur & Abfragen
- Power BI → Dashboard

Gesamtarchitektur

Der komplette Datenfluss beginnt bei der Eingabe über Streamlit. Die Daten werden als CSV gespeichert, in AWS S3 abgelegt, über Glue und Athena strukturiert und anschließend in Power BI analysiert. Ziel dieser Architektur ist ein **durchgängiger Datenworkflow**, von der Datenerfassung bis zur professionellen Auswertung

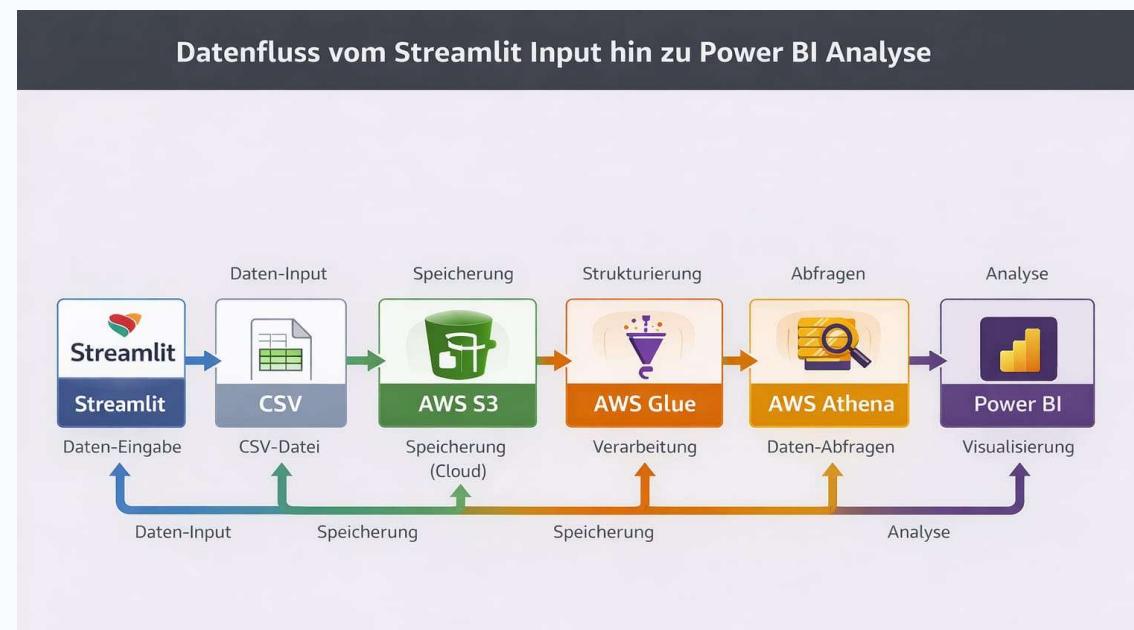
Warum habe ich mich für diese Datenarchitektur entschieden?

Klare Trennung der Aufgaben: Streamlit erfasst Daten, S3 speichert sie, Glue strukturiert sie, Athena analysiert sie, Power BI visualisiert sie.

Einfach & praxisnah: Keine komplexe Datenbank nötig, trotzdem SQL-Analysen möglich.

Skalierbar & erweiterbar: Die Architektur funktioniert für kleine Projekte und lässt sich später problemlos ausbauen.

Realitätsnah: Entspricht typischen Cloud-Data-Pipelines aus der Praxis.



Power Query & Datenbereinigung

- Rohdaten aus CSV (AWS S3 / Athena)
- Daten nicht sofort analysefähig für Power BI
- Unterschiedliche Formate und Schreibweisen

Ausgangstabelle: S3_Zeiterfassung

→ zentrale **Faktentabelle** mit allen Rohdaten (Datum, Kursname, Lernart, Zeiten, Dauer)

⚠ Aufgetretene Probleme:

- Inkonsistente Textwerte z. B. „Projekt“ vs. „PROJEKT“
- Zeitspalten im Textformat (z. B. „10:30“)
- Redundante Spalten
- Dezimalzahlen im deutschen Format (Komma statt Punkt)
- Fehlerhafte Einträge (NULL für KursID Werte im Oktober)
- Doppelte Einträge

✓ Lösungen in Power Query

- Textbereinigung (Groß-/Kleinschreibung vereinheitlicht)
- Umwandlung von Text → Datum / Uhrzeit
- Duplikaten entfernen

Power Query & Datenmodellierung (1)

- Keine redundanten Textwerte
- Konsistente Schreibweise
- **Fakten messen, Dimensionen beschreiben**

Dimensionen „Lernart“ und „Kurs“ erstellen

- Tabelle duplizieren: S3_Zeiterfassung
- Umbenennen: Dim_Lernart/Dim_Kurs
- Spaltenreduktion : Alle Spalten löschen außer Lernart/Kurs
- Duplikate entfernen
- ID erstellen

 Ergebnis: saubere Dimensionstabelle Lernarten und Kursen

Faktentabelle normalisieren

- In S3_Zeiterfassung: Join (Merge Queries) mit:
 - Dim_Lernart → über Lernart
 - Dim_Kurs → über Kursname

Nach dem Join:

Nur die ID-Spalten behalten
Textspalten (Lernart, Kursname) entfernen

 Faktentabelle enthält nur noch Fremdschlüssel

Power Query & Datenmodellierung (2)

- Keine redundanten Textwerte
- Konsistente Schreibweise
- **Fakten messen, Dimensionen beschreiben**

datum	wochentag	startzeit	endzeit	dauer (h)	lernart_ID	kursname_ID
Freitag, 12. Dezember 2025	Freitag	15:30	17:30	2,0	2	11
Freitag, 12. Dezember 2025	Freitag	13:30	14:30	1,0	4	17
Donnerstag, 11. Dezember 2025	Donnerstag	12:00	15:00	3,0	2	11
Dienstag, 9. Dezember 2025	Dienstag	10:15	12:15	2,0	2	8
Montag, 8. Dezember 2025	Montag	18:30	19:15	0,8	3	8
Montag, 8. Dezember 2025	Montag	12:30	13:30	1,0	3	8
Montag, 8. Dezember 2025	Montag	10:00	11:15	1,3	3	8
Sonntag, 7. Dezember 2025	Sonntag	18:30	21:45	3,3	2	10
Samstag, 6. Dezember 2025	Samstag	12:15	13:00	0,8	2	10
Samstag, 6. Dezember 2025	Samstag	12:00	12:15	0,3	2	10
Freitag, 5. Dezember 2025	Freitag	18:30	21:30	3,0	2	10
Donnerstag, 4. Dezember 2025	Donnerstag	11:00	14:00	3,0	2	10

lernart	lernart_ID
	1
Selbststudium	2
Theorie (Video)	3
Live-Call-Vorlesung	4
Live-Call-Hausaufgaben	5

kursname	kursname_ID
Excel	1
Teilprojekt	2
Machine Learning	3
Python	4
PowerBI	5
SQL	6
ETL	7
BigData	8
Streamlit	10
PROJEKT	11
Karrierecoaching	12
AWS	13
ScrumPO	14
EDA	15
Statistik	16
Organisation	17
Tableau	18
Docker	19
NoSQL	20
GitHub	21

- Saubere Struktur
- Korrekte Sortierung
- Bessere Auswertungen

Datum- und Zeit_Dimensionstabellen

Dimensionstabellen sorgen für Ordnung. Zeitinformationen werden nur einmal definiert, korrekt sortiert und können flexibel für Auswertungen genutzt werden

Warum eine Dim_Datum-Tabelle?

- Ermöglicht Auswertungen nach:
Jahr, Monat, Kalenderwoche
Wochentag (korrekt sortiert, nicht alphabetisch)
- Einheitliche Zeitlogik für alle Berichte
- Vermeidet doppelte Berechnungen in jeder Visualisierung
- Datum ist mehr als nur ein Wert – es ist eine Analyse-Dimension**

Date	Jahr	Monat_Nr	KW	Wochentag	Monat	Jahr_Monat	JahrMonat_Sort	Wochentag_Nr
Dienstag, 27. Januar 2026	2026	1	5	Montag	Januar	2026-01	202601	1
Sonntag, 25. Januar 2026	2026	1	5	Sonntag	Januar	2026-01	202601	7
Samstag, 24. Januar 2026	2026	1	4	Samstag	Januar	2026-01	202601	6
Freitag, 23. Januar 2026	2026	1	4	Freitag	Januar	2026-01	202601	5
Donnerstag, 22. Januar 2026	2026	1	4	Donnerstag	Januar	2026-01	202601	4
Mittwoch, 21. Januar 2026	2026	1	4	Mittwoch	Januar	2026-01	202601	3
Dienstag, 20. Januar 2026	2026	1	4	Dienstag	Januar	2026-01	202601	2
Montag, 19. Januar 2026	2026	1	4	Montag	Januar	2026-01	202601	1
Sonntag, 18. Januar 2026	2026	1	4	Sonntag	Januar	2026-01	202601	7
Samstag, 17. Januar 2026	2026	1	3	Samstag	Januar	2026-01	202601	6
Freitag, 16. Januar 2026	2026	1	3	Freitag	Januar	2026-01	202601	5

Warum eine Dim_Zeit-Tabelle?

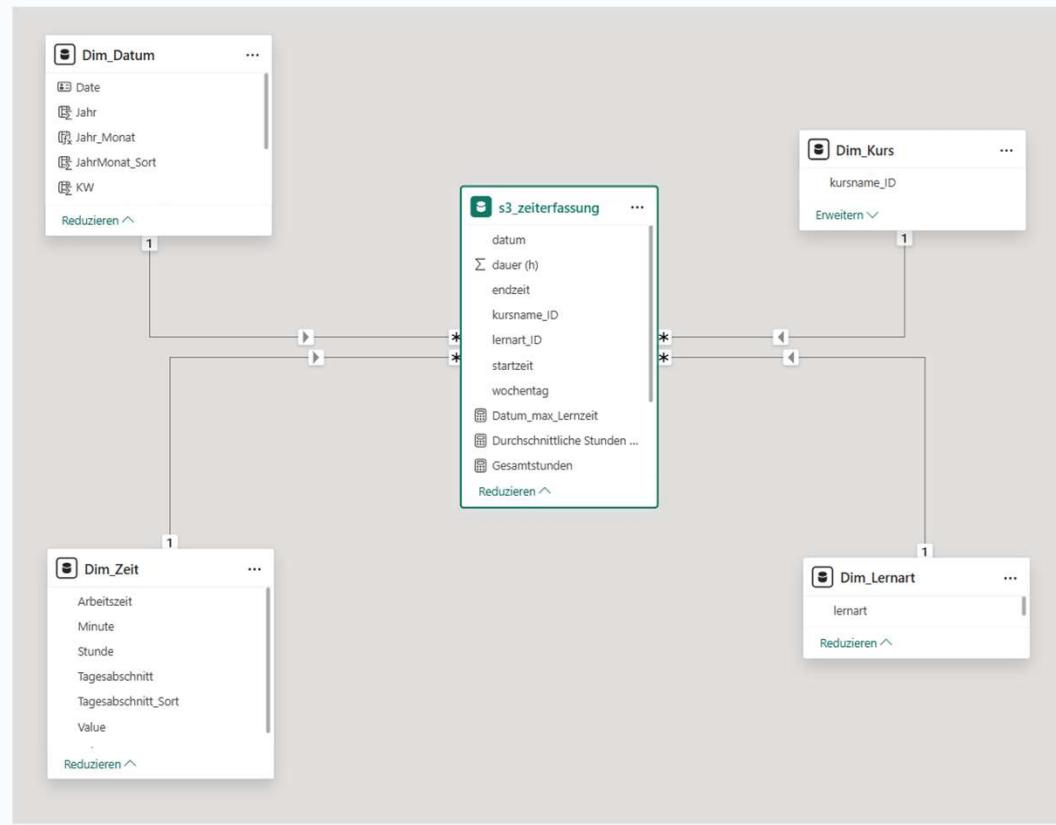
- Ermöglicht:
Analyse nach Stunde, Minute
Tagesabschnitten (Nacht, Früher Morgen, Morgen,
Nachmittag, Abend)
- Einheitliche Definition von Zeitlogik an einer Stelle
- Zeit braucht Kontext, nicht nur ein Uhrzeitfeld**

Value	Stunde	Minute	Zeit	Tagesabschnitt	Tagesabschnitt_Sort	Arbeitszeit
6	6	0	06:00:00	Früher Morgen 06:09	2	False
6,25	6	15	06:15:00	Früher Morgen 06:09	2	False
6,5	6	30	06:30:00	Früher Morgen 06:09	2	False
6,75	6	45	06:45:00	Früher Morgen 06:09	2	False
7	7	0	07:00:00	Früher Morgen 06:09	2	False
7,25	7	15	07:15:00	Früher Morgen 06:09	2	False
7,5	7	30	07:30:00	Früher Morgen 06:09	2	False
7,75	7	45	07:45:00	Früher Morgen 06:09	2	False
8	8	0	08:00:00	Früher Morgen 06:09	2	False
8,25	8	15	08:15:00	Früher Morgen 06:09	2	False
8,5	8	30	08:30:00	Früher Morgen 06:09	2	False
8,75	8	45	08:45:00	Früher Morgen 06:09	2	False
9	9	0	09:00:00	Vormittag 09:12	3	True

Datenmodell

- Fact_Zeiterfassung
- Dim_Kurs
- Dim_Lernart
- Dim_Datum
- Dim_Zeit
- Star Schema

Für die Analyse habe ich ein sogenanntes Star Schema verwendet. Dabei gibt es eine Faktentabelle mit den Lernstunden und Dimensionstabellen für Kurs, Lernart, Datum und Zeit



- Zeiteinträge können mehrere Tagesabschnitte überschreiten
- Tag wird in 15-Minuten-Slots aufgeteilt
- Jeder Slot wird geprüft, ob er zum Zeiteintrag gehört.
- Die Stunden werden anteilig den Tagesabschnitten zugeordnet

Zeitliche Aufteilung von Arbeitsstunden in Tagesabschnitte

Value	Stunde	Minute	Zeit	Tagesabschnitt	Tagesabschnitt_Sort	Arbeitszeit
6	6	0	06:00:00	Früher Morgen 06:09	2	False
6,25	6	15	06:15:00	Früher Morgen 06:09	2	False
6,5	6	30	06:30:00	Früher Morgen 06:09	2	False
6,75	6	45	06:45:00	Früher Morgen 06:09	2	False
7	7	0	07:00:00	Früher Morgen 06:09	2	False
7,25	7	15	07:15:00	Früher Morgen 06:09	2	False
7,5	7	30	07:30:00	Früher Morgen 06:09	2	False
7,75	7	45	07:45:00	Früher Morgen 06:09	2	False
8	8	0	08:00:00	Früher Morgen 06:09	2	False
8,25	8	15	08:15:00	Früher Morgen 06:09	2	False

Fragestellung

Wie viele Arbeitsstunden fallen in welchen Tagesabschnitt?

Beispiel:	Zeiteintrag	Tagesabschnitt	Stunden
	05:00 – 10:00	Nacht (00–06)	1,00 h
	05:00 – 10:00	Früher Morgen (06–09)	3,00 h
	05:00 – 10:00	Vormittag (09–12)	1,00 h

Ein Zeiteintrag kann mehrere Tagesabschnitte überlappen.

Wir müssen also prüfen, welcher Teil der Zeit in welchen Abschnitt fällt.

Der Tag wird in **96 Slots à 15 Minuten** aufgeteilt (00:00–00:15, 00:15–00:30, ..., 23:45–24:00)

Jeder Slot gehört **genau einem Tagesabschnitt**

Beispiel: 05:00–10:00 → wird in **20 Slots** unterteilt:

05–06 Uhr → Nacht

$$\rightarrow 4 \text{ Slots} \times 0,25 \text{ h} = 1 \text{ h}$$

06–09 Uhr → Früher Morgen

$$\rightarrow 12 \text{ Slots} \times 0,25 \text{ h} = 3 \text{ h}$$

09–10 Uhr → Vormittag

$$\rightarrow 4 \text{ Slots} \times 0,25 \text{ h} = 1 \text{ h}$$

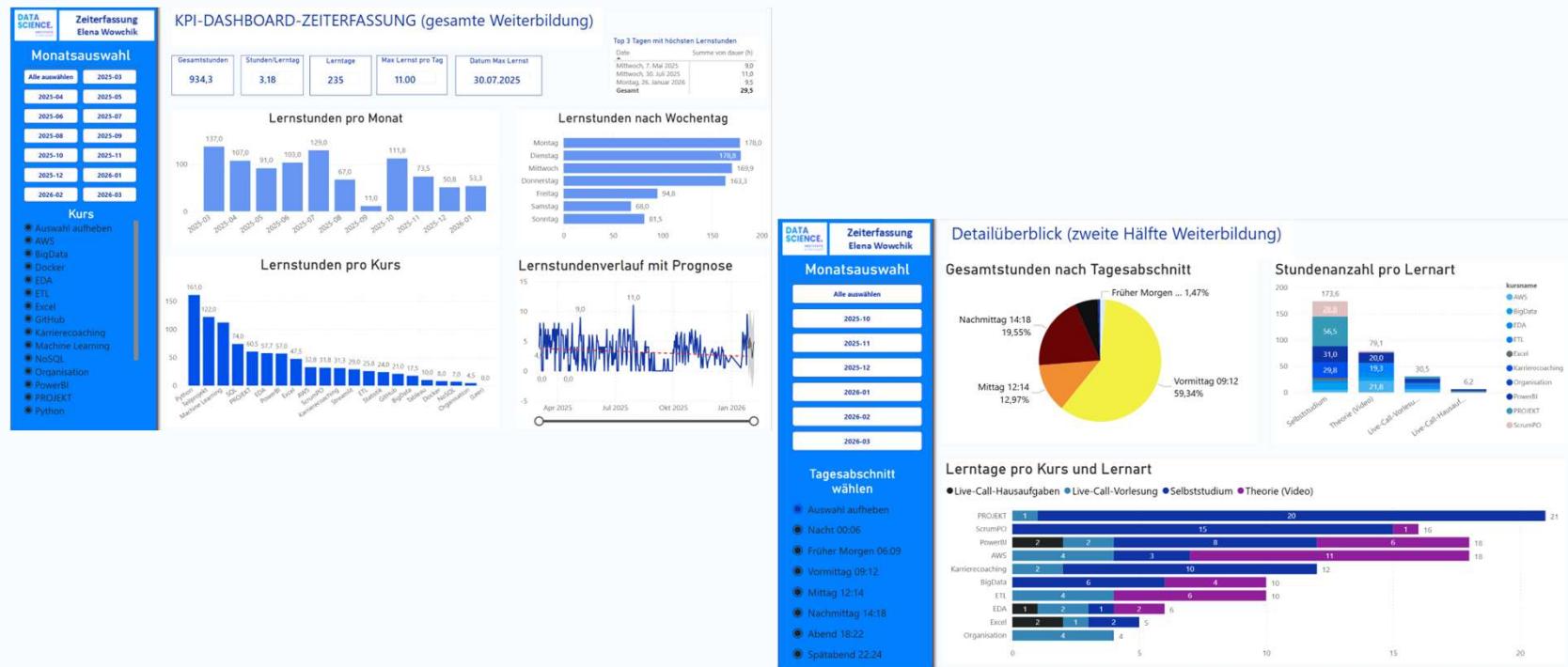
Für jeden Slot wird geprüft, ob der innerhalb des Zeiteintrags liegt:

wenn ja → 0,25 St. zählen, wenn nein —0 St. zählen

Power BI

- Dashboards & KPIs
- Zeitliche Analysen
- Top-Lerntage

In Power BI verbinde ich mich mit Athena und erstelle Dashboards. Hier werden die Lernzeiten visuell dargestellt, zum Beispiel nach Tagen, Monaten oder Tagesabschnitten



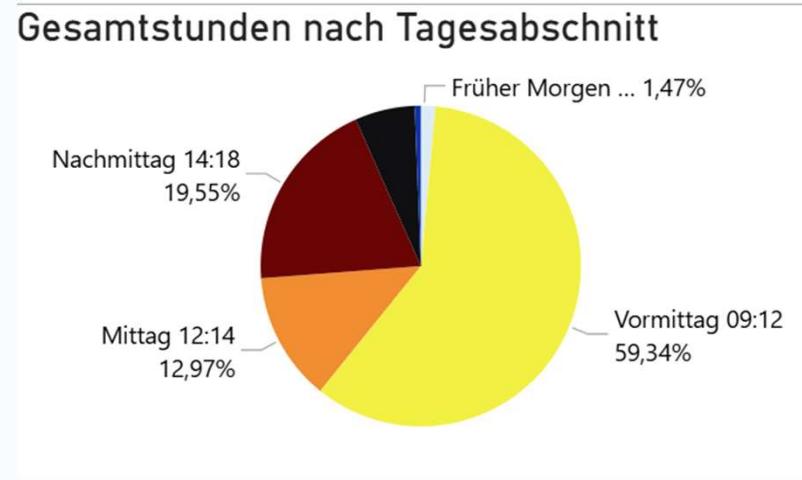
- Gesamtlernzeit
- Durchschnitt pro Tag
- Längster Lerntag
- Tagesabschnitte

KPI-Dashboard

Im KPI-Dashboard sehe ich auf einen Blick die wichtigsten Kennzahlen, zum Beispiel meine Gesamtlernzeit oder den längsten Lerntag

KPI-DASHBOARD-ZEITERFASSUNG (gesamte Weiterbildung)

Gesamtstunden	934,3	Stunden/Lerntag	3,18	Lerntage	235	Max Lernst pro Tag	11.00	Datum Max Lernst	30.07.2025
---------------	-------	-----------------	------	----------	-----	--------------------	-------	------------------	------------

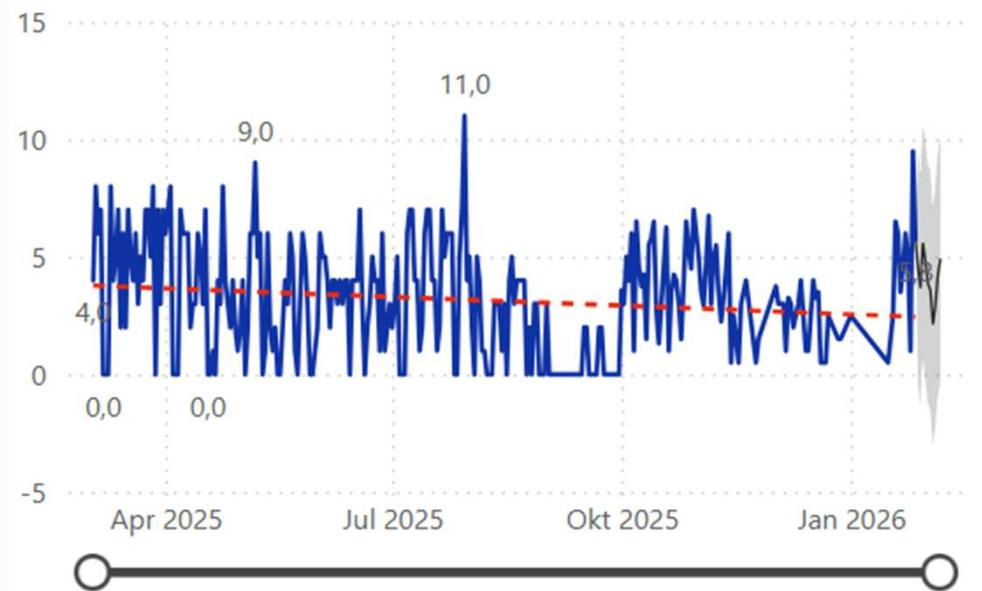


- Produktivste Tage
- Lernverhalten
- Zeitliche Muster

Erkenntnisse

Durch die Analyse erkenne ich Muster: An welchen Tagen ich am produktivsten bin, zu welchen Uhrzeiten ich lerne und wie sich meine Lernzeiten über Monate entwickeln

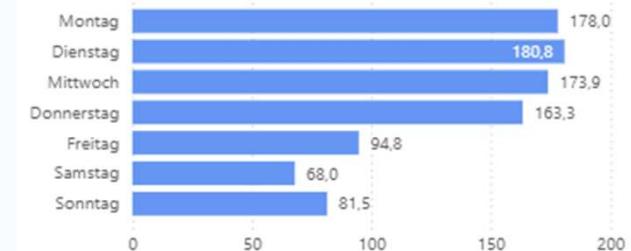
Lernstundenverlauf mit Prognose



Top 3 Tagen mit höchsten Lernstunden

Date	Summe von dauer (h)
Mittwoch, 7. Mai 2025	9,0
Mittwoch, 30. Juli 2025	11,0
Montag, 26. Januar 2026	9,5
Gesamt	29,5

Lernstunden nach Wochentag



- End-to-End BI-Projekt
- Cloud + Analyse
- Praxisnah

Ich habe ein vollständiges BI-Projekt umgesetzt – von der Datenerfassung bis zum Dashboard. Dabei habe ich ETL, Cloud-Technologien, Datenmodellierung und Analyse praktisch angewendet

Fazit

- Mehr Nutzer
- Mehr KPIs
- Automatisierung

Ausblick

Das Projekt könnte leicht erweitert werden, zum Beispiel um die Unterstützung mehrerer Nutzer oder zusätzliche Filtermöglichkeiten (Themen, Dozentname usw). Die aktuelle Streamlit-Eingabemaske ist eher einfach gehalten und könnte noch benutzerfreundlicher gestaltet werden.

Das Ziel des Projekts war nicht die Erstellung eines fertigen Dashboards und Eingabemaske, sondern vor allem das **Verständnis des gesamten Datenprozesses**, von der Aufbereitung über Analyse bis zur Visualisierung

Vielen Dank für Ihre Aufmerksamkeit!

GitHub: github.com/Elena-Wowchik/Projekt_Zeiterfassung

LinkedIn: linkedin.com/in/Elena-Wowchik

E-Mail: elena.wowchik@gmx.de