Project Name	...
Online team meeting	https://fau.zoom-x.de/j/61181981845?pwd=ZJq0iXvp6t5yTNBjRccLMqrUQEylWF.1
Production system (if any)	...
Test system (if any)	...
GitHub repository	https://github.com/amosproj/amos2024ws02-backup-metadata-analyzer
GitHub feature board	https://github.com/orgs/amosproj/projects/71/views/2
GitHub imp-squared backlog	https://github.com/orgs/amosproj/projects/75/views/1
Team T-shirt female (white)	https://www.shirtinator.de/s/scY-JglCTmu_7ZBCkeWx0w
Team T-shirt female (black)	https://www.shirtinator.de/s/zM5PRDBVRdOw03ch6thh4w
Team T-shirt male (white)	https://www.shirtinator.de/s/DUbamNAcRXukdXcISNchvA
Team T-shirt male (black)	https://www.shirtinator.de/s/97GiPyFdTre3YHtD_MIpIQ
Additional materials	...
Team maling list	oss-amos-proj2@lists.fau.de

Last Name	First Name	GitHub User Name	Email Address
Sulzbach	Lara	LaraSlzb	lara.sulzbach@fau.de
Oberndörfer	Florian	flo0852	florian8751t@gmail.com
Schnell	Oliver	Omega65536	mail@oliver-schnell.de
Klingenberg	Christoph	chrisklg	christoph.klingenberg@fau.de
Regl	Amelie	heskil	mellyre42@gmail.com
Engelhard	Dirk	engelharddirk	dirk.engelhard@hotmail.de
Rauen	Moritz	LHMoritz	moritzrauen.mr@gmail.com
Garbe	Valentin	Us3rname11	garbevalentin@gmail.com
Deli	Deniz	ddeli	d.deli@campus.tu-berlin.de

#	Meeting Day	Product Owners	Software Developer	Release Manager	Scrum Master	Comment
1	2024-10-16	Moritz Rauen	Everyone else	N/A	Lara Sulzbach	
2	2024-10-23	Valentin Garbe	Everyone else	Amelie Regl	Lara Sulzbach	
3	2024-10-30	Moritz Rauen	Everyone else	Florian Oberndörfer	Lara Sulzbach	
4	2024-11-06	Valentin Garbe	Everyone else	Oliver Schnell	Lara Sulzbach	
5	2024-11-13	Moritz Rauen	Everyone else	Dirk Engelhard	Lara Sulzbach	
6	2024-11-20	Valentin Garbe	Everyone else	Christoph Klingenberg	Lara Sulzbach	
7	2024-11-27	Moritz Rauen	Everyone else	Deniz Deli	Lara Sulzbach	Mid-term due
8	2024-12-04	Valentin Garbe	Everyone else	Amelie Regl	Lara Sulzbach	
9	2024-12-11	Moritz Rauen	Everyone else	Florian Oberndörfer	Lara Sulzbach	
10	2023-01-08	Valentin Garbe	Everyone else	Oliver Schnell	Lara Sulzbach	
11	2023-01-15	Moritz Rauen	Everyone else	Dirk Engelhard	Lara Sulzbach	
12	2023-01-22	Valentin Garbe	Everyone else	Christoph Klingenberg	Lara Sulzbach	
13	2023-01-29	Moritz Rauen	Everyone else	Deniz Deli	Lara Sulzbach	
14	2023-02-05	Valentin Garbe	Everyone else	Amelie Regl	Lara Sulzbach	Demo day!
15	2023-02-12	Moritz Rauen	Everyone else	Florian Oberndörfer	Lara Sulzbach	Retrospective
Product owners, software developers, and Scurm Master are set and ideally don't change over time; the critical part is the Release Manager role you need to define here						

Goals	Be respectful to each other
	Have fun together
	Learn new things, improve software engineering, team-working skills
	create valuable outcome
	Help each other
Meeting norms	Be on time
	Inform the team about unavalability
	Keep it short and conciese
Working norms	Ask for help, support each other
	follow coding best-practices: work on feature-branch, comment code, commit often
Coordination norms	SD can assign themselves to tickets in the sprint backlog
	PO's are moderating the team meeting
	Every new code must be reviewed before merge
	Do technical reviews before assigning to a new issue
Communication norms	Communication is mainly over Slack
	internal communication in German, external in English
	Honestly over performance, do not try to cover mistakes
Consideration norms	PO's should lead the discussion
Cont. improvement norms	Give directly adressed and constructive feedback
Rewards	
Sanctions	If you are not in the team meeting at 12.35 without excuse, SD's have to do an extra review, PO's have to take over the next sprint moderating
Signatures	
Scrum Master	Lara Sulzbach
Product owner	Valentin Garbe
Product owner	Moritz Rauen
Software developer	Oliver Schnell
Software developer	Amelie Regl
Software developer	Florian Oberndörfer
Software developer	Christoph Klingenberg
Software developer	Dirk Engelhard
Software developer	Deniz Deli

Product Vision	Project Mission
Databackups are an essential functionality for businesses. They ensure, that no important information or artifacts of work is lost due to technical errors or hacking. This metadata analizer aims to help detect problems or anomalies with backups. This problem detection is done in a timely manor, and in some cases even predictively, to help the customer to react before major damages occur.	This project aims to explore which data analysis methods are applicable and useful to for the metadata, which is generated by the industry partner´s software "SESAM". Furthermore, the results of the analysis are condensed into insights and displayed with graphs and alerts. This will help the customer to better understand their backup data and focus on the important take aways.

Term	Definition
Analyzer	Our Python Backen Modul which handles the AI analyzer
Backend	Our Main Backend Module which connects Frontent, Database and Analyzer
Monorepo	A monorepo is a single repository housing code for multiple projects

Sprint #	Sprint goal
1	None
2	None
3	None
4	Understanding the customers datastructure and analysis methods for them
5	Real Data Flow
6	**Analysis Method Implementation**
7	Density based & Time Series Analysis implemented
8	
9	
10	
11	
12	
13	
14	
15	

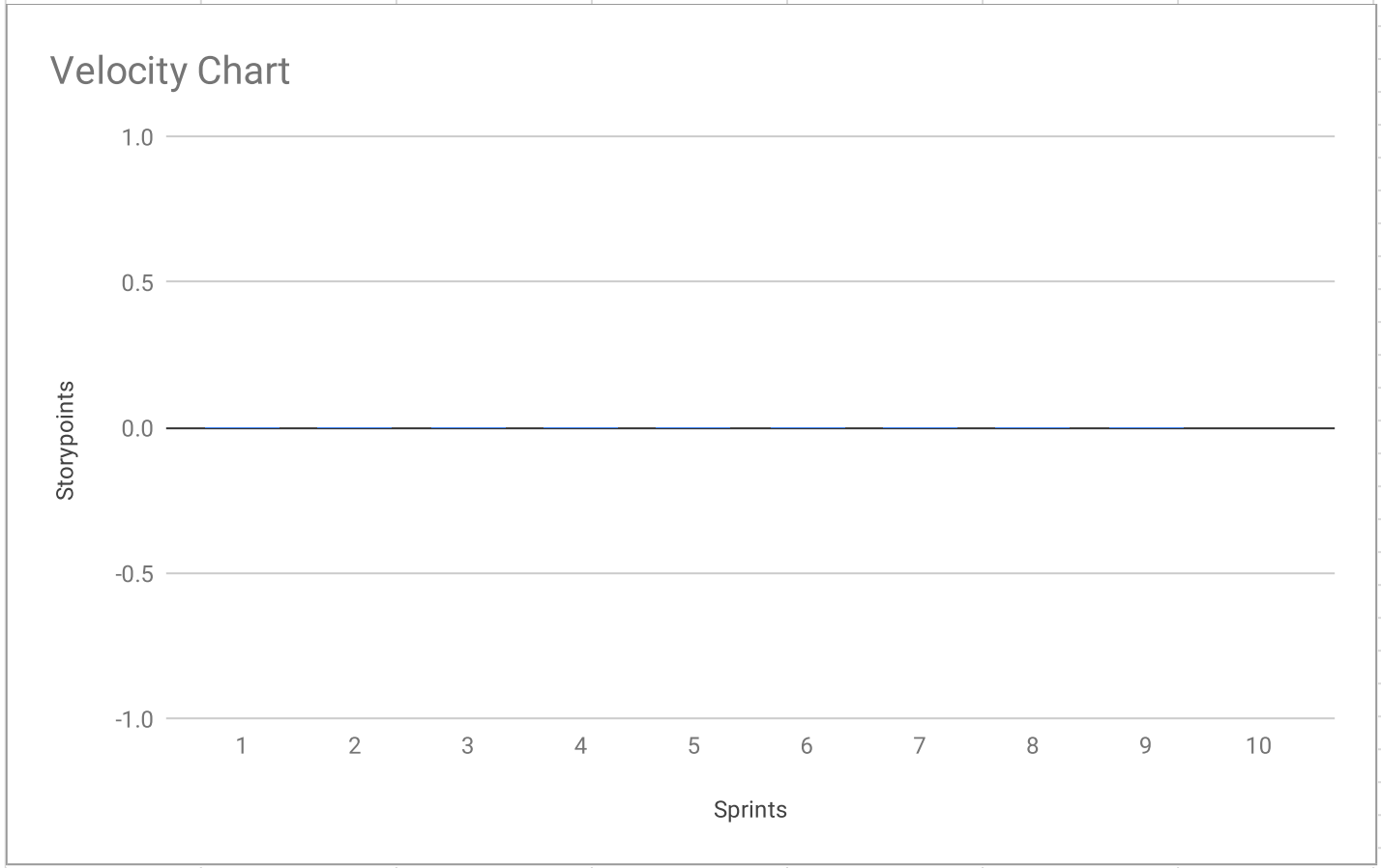
Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
Release						
Total			111	111		
Sprints						
1	**Dev Environment Setup**		3	111	2	111
2	**Architecture Foundation**		15	108	19	109
3	**Build Process Setup**		11	93	12	90
4	**Understanding the customers datastructure and analysis methods for them**		29	82	26	78
5	**Real Data workflow**		36	53	11	52
6	**Analysis Method Implementation**		17	17	0	41
Features						
1	**Dev Environment S**	Dev Environment Setup	3		2	
2	**Architecture Foun**	Landing page with backup data table	2		2	
		Initialize Frontend module	2		2	
		Create Architecture Diagram	1		1	
		Architecture evaluation and decision	3		2	
		Initialize Backend Module	2		8	
		Model analysis research	3		2	
		Concept of build process defined & local build implemented	2		2	
3	**Build Process Setup**	Build Process Review preperation	1		2	
		Implement build process for release	8		8	
		Decide on Build process steps for release	2		2	

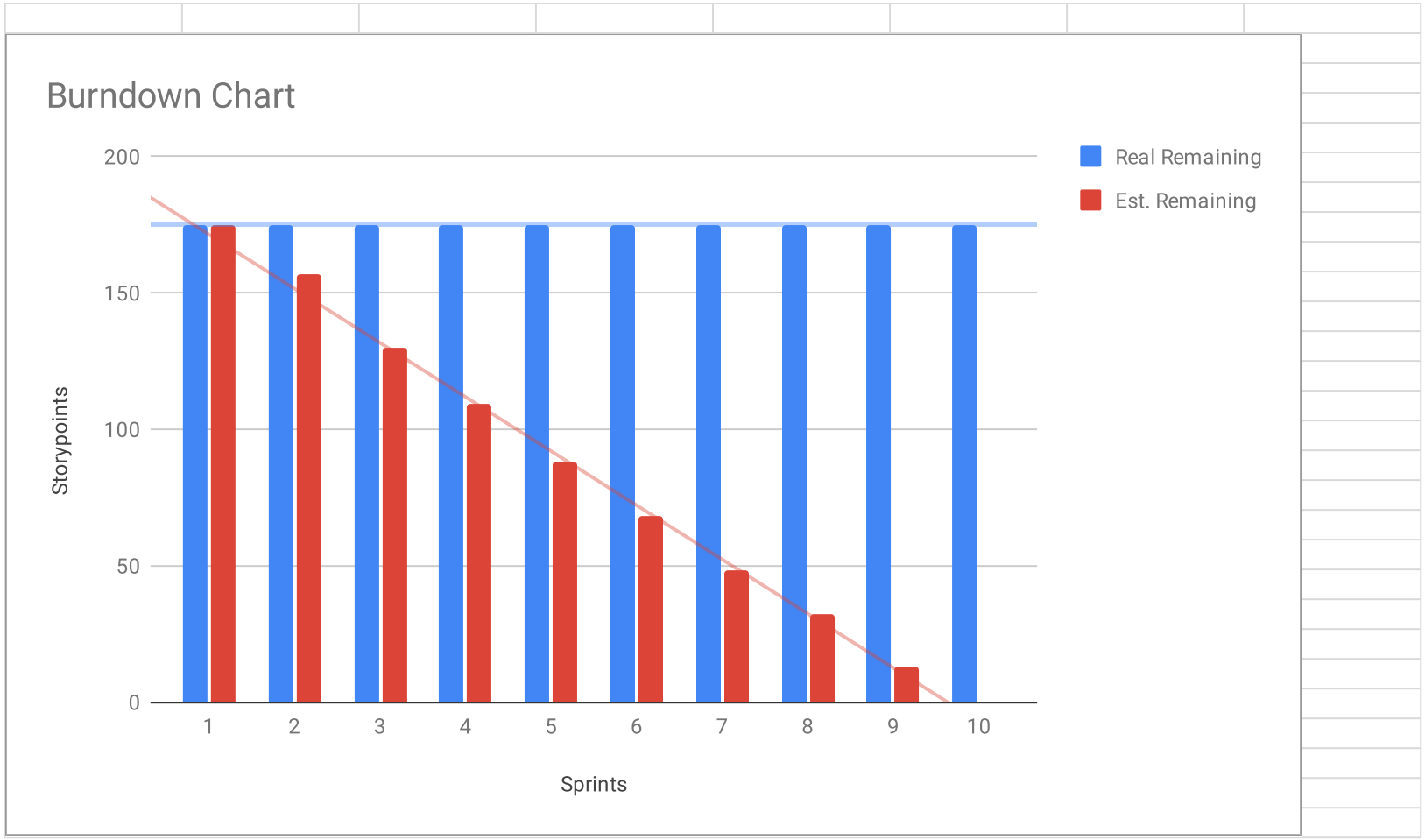
Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
4	**Understanding the customers datastructure and analysis methods for them**	Am Charts integration	2		2	
		First Test with Analysis module	3		3	
		Init the Analysis module	3		3	
		Analysis Methods: Rule Based Analysis	3		3	
		Analysis Methods: Density Based Analysis	5		5	
		Analysis Methods: Time Series Analysis	5		5	
		Analysis Methods: Neural Networks	8		5	
5	Real Data workflow	Being familiar with the Data	3		3	
		Backend: Email Notification Trigger	3		3	
		Unit & Integration Test Implementation Backend	3		3	
		Homework: Record and upload build process video	2		2	
6	Analysis Method Implementation	Github Actions: Test Automation	5			
		Frontend: Blueprint Alert Panel	5			
		Analysis Module: First Rule Based Analysis Implemented	5			
		Build Documentation	2			
		Unit & Integration Test Implementation Frontend	3			
		Backend: Backup Size-Timeline API	5			
		Frontend: Display Backup Size-Timeline	3			
		Backend: Alert Handling	3			
		Improve local setup - Containerization: Analyzer Module + Database	8			

Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
Release						
Total			175	175		
Sprints						
1	Density based & Time Series Analysis implemented		18	175	0	175
2	Density based & Time Series Analysis Improved		27	157	0	175
3	Machine Learning Analysis Implemented		21	130	0	175
4	Anomaly detection implemented		21	109	0	175
5	Improvement of Analysis		20	88	0	175
6	Improvement of Analysis & Cleanup		20	68	0	175
7	Demo Day Preparations		16	48	0	175
8	Final Release		19	32	0	175
9	Report Created		13	13	0	175
10				0		175
Features						
1		Analysis Module: First Density Based Analysis Implemented	5			
		Analysis Module: First Time Series Analysis Implemented	5			
		Secure API Endpoints	5			
		Analysis Module: Further Rule Based Analysis implemented	3			
2		Analysis Module: Improved Density Based Analysis implemented	8			
		Analysis Module: Improved Time Series Analysis implemented	8			
		Frontend: Advanced Alerts	3			
		Analysis Module: Machine Learning Analysis first tests	8			
3		Analysis Module: Machine learning Analysis implemented	8			
		Analysis Module: Improved Time Series Analysis improved	8			
		Analysis Module: Improved Density Based Analysis improved	5			
4		Anomaly Detection in Backup History Implemented	8			
		Outlier Detection in Backup Implemented	8			
		Frontend: Anomaly visualisation	5			

Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
5		Improvement of Analysis capabilities: Density Based	5			
		Improvement of Analysis capabilities: Time Series Based	5			
		Improvement of Analysis capabilities: Machine Learning	5			
		Frontend: Visualisation & Workflow Improvements	5			
6		Improvement of Analysis capabilities: Density Based	5			
		Improvement of Analysis capabilities: Time Series Based	5			
		Improvement of Analysis capabilities: Machine Learning	5			
		Frontend: Visualisation & Workflow Improvements	5			
7		Documentation Cleanup	3			
		Code Cleanup	5			
		Preparing Demo Material (Slides & Video)	3			
		Bugfixing	5			
8		Bugfixing	8			
		Build well tested	8			
		Demo Prepared	3			
9		Report written	13			

Average Verlocity	0	storypoints / sprint




Burndown Chart
200
150
100
50
0
Storypoints
1
2
3
4
5
6
7
8
9
10
Sprints
Real Remaining
Est. Remaining

#	Feature Definition of Done	Sprint Release Definition of Done	Project Release Definition of Done
	Github actions pipline: success / All test: success	Sprint board is cleaned up	Well tested software
	All acceptance criteria are met	Release candidate is demoed and approved	No "loose ends" in the codebase
	Code review done and approved	Release Dev branch is merged to main	Stable running software
	Is merged to Dev branch		Documentation is complete
	Tests for new feature are implemented		Industry partners have no questions left about functionality and usage
			Industry partners are satisfied with value of the product

Type	Link / reference

#	Context	Name	Version	License	Comment
1	Frontend	Angular Devkit Build Angular	18.2.0	MIT	
2	Frontend	Angular Devkit Core	18.2.0	MIT	
3	Frontend	Angular Devkit SChematics	18.2.0	MIT	
4	Frontend	Angular CLI	18.2.0	MIT	
5	Frontend	Angular Compiler CLI	18.2.0	MIT	
6	Frontend	Angular Language Service	18.2.0	MIT	
7	Linter	Eslint JS	9.8.0	MIT	
8	Technical setup	NestJS Schematics	10.0.1	MIT	
9	Technical setup	Nest JS Testing	10.0.2	MIT	
10	Technical setup	Nx Angular	20.0.5	MIT	
11	Technical setup	Nx Cypress	20.0.5	MIT	
12	Technical setup	Nx EsBuild	20.0.5	MIT	
13	Linter	Nx EsLint	20.0.5	MIT	
14	Linter	Nx EsLint Plugin	20.0.5	MIT	
15	Technical setup	Nx Jest	20.0.5	MIT	
16	Technical setup	Nx js	20.0.5	MIT	
17	NX App Generator	Nx Nest	20.0.5	MIT	
18	NX App Generator	Nx Node	20.0.5	MIT	
19	NX App Generator	Nx web	20.0.5	MIT	
20	Technical setup	Schematics Angular	18.2.0	MIT	
21	Technical setup	SWC-Node Register	1.9.1	MIT	
22	Technical setup	SWC-Core	1.5.7	Apache-2.0	
23	Technical setup	SWC-Helpers	0.5.11	Apache-2.0	
24	Types	Types Cors	2.8.17	MIT	
25	Types	Types Jest	29.5.12	MIT	
26	Types	Types Node	18.16.9	MIT	
27	Linting	Typescript-eslint Utils	8.0.0	MIT	
28	Linting	Angular EsLint	18.3.0	MIT	
29	Testing Framework for E2E and Component Tests	Cypress	13.13.0	MIT	
30	Builder	EsBuild	0.19.2	MIT	
31	Make decorators useable with EsBuild	EsBuild Plugin Tsc	0.4.0	MIT	
32	Linter	EsLint	9.8.0	MIT	
33	Prettier Config for EsLint	EsLint Config Prettier	9.0.0	MIT	
34	EsLint Plugin for Linting in Cypress	EsLint Plugin Cypress	3.5.0	MIT	
35	Tests	Jest	29.7.0	MIT	
36	Tests	Jest Environment JSDom	29.7.0	MIT	

#	Context	Name	Version	License	Comment
37	Tests	Jest Environment node	29.7.0	MIT	
38	Tests	Jest Preset Angular	14.1.0	MIT	
39	Routing	JS Dom	22.1.0	MIT	
40	Monorepo setup	Nx	20.0.5	MIT	
41	Formatter	Prettier	2.6.2	MIT	
42	Testing	ts-jest	29.1.0	MIT	
43	Transpiler	ts-node	10.9.1	MIT	
44	Technical setup	tslib	2.3.0	0BSD	
45	Language for Frontend and Backend	typescript	5.5.2	Apache-2.0	
46	Linter for Typescript	typescript EsLint	8.0.0	MIT	
47	Frontend	Angular Animations	18.2.0	MIT	
48	Frontend	Angular Common	18.2.0	MIT	
49	Frontend	Angular Compiler	18.2.0	MIT	
50	Frontend	Angular Core	18.2.0	MIT	
51	Frontend	Angular Forms	18.2.0	MIT	
52	Frontend	Angular Platform Browser	18.2.0	MIT	
53	Frontend	Angular Platform Browser Dynamic	18.2.0	MIT	
54	Frontend	Angular Router	18.2.0	MIT	
55	Frontend Components	CDS Core	6.14.0	MIT	
56	CSS Framework + Components	CLR Angular	17.4.0	MIT + SIL Open Font License version 1.1	SIL Open Font License version 1.1 only for Fonts
57	CSS Framework + Components	CLR UI	17.3.1	MIT + SIL Open Font License version 1.1	SIL Open Font License version 1.1 only for Fonts
58	Backend Components	Nest JS Common	10.0.2	MIT	
59	Backend Components	Nest JS Config	3.3.0	MIT	
60	Backend Components	Nest JS Core	10.0.2	MIT	
61	Backend Components	Nest JS Platform Express	10.0.2	MIT	
62	API Docs	Nest JS Swagger	7.4.2	MIT	
63	Nest JS Database Support	Nest JS Typeorm	10.0.2	MIT	
64	Promise-based HTTP Client for Node	Axios	1.6.0	MIT	
65	transform plain object to some instance of class	Class Transformer	0.5.1	MIT	
66	Validations (e.g. for Dtos)	Class Validator	0.14.1	MIT	
67	Cross Origin Requests	Cors	2.8.5	MIT	
70	PostgreSQL Client for Node JS	pg	8.13.1	MIT	

#	Context	Name	Version	License	Comment
71	Runtime Reflections for types	Reflect-Metadata	0.1.13	Apache-2.0	
72	Reactive Programming (e.g. Observables)	RXJS	7.8.0	Apache-2.0	
73	API Docs UI	Swagger UI Express	5.0.1	MIT	
74	ORM	Typeorm	0.3.20	MIT	
75	Frontend	Zone JS	0.14.3	MIT	
76	Python	markupsafe	3.0.2	BSD	
77	Python	blinker	1.8.2	MIT	
78	Python	click	8.1.7	BSD	
79	Python	itsdangerous	2.2.0	BSD	
80	Python	jinja2	3.1.4	BSD	
81	Python	werkzeug	3.1.1	BSD	
82	Python	flask	3.0.3	BSD	
83	HttpModule for connecting to Analyzer Service	NestJS Axios	3.1.1	MIT	
84	Python to set environment variables	python-dotenv	1.0.1	BSD	
85	Frontend Chart-Design	amcharts5	5.10.7		
86	Frontend Chart-Design	amCharts 5 fonts	5.0.1		
87	Python Testing	Pytest	7.3.1	MIT	
88	Test Framework for Backend	NRWL Jest	19.8.4	MIT	
89	SQLAlchemy	SQL toolkit for Python	2.0.36	MIT	
90	pg8000	Postgres interface Library	1.31.2	MIT	
91	python-dateutil	extension to the standard Python datetime module	2.9.0	BSD	
92	asn1crypto	fast ASN.1 parser	1.5.1	MIT	
93	scramp	python implementation of SCRAM protocol	1.4.5	MIT	
94	Mock API Calls	Supertest	7.0.0	MIT	
95	Mock API Calls	Types Supertest	6.0.2	MIT	
96	Mailing	Nodemailer	6.4.16	MIT-0	
97	Mailing	@nestjs-modules/mailer	2.0.2	MIT	
98	Template Builder for Mailing	handlebars	4.7.8	MIT	
99	Mailing	nodemailer	6.9.16	MIT-0	
100	Swagger UI for Python	flasgger	0.9.7.1	MIT	
101	Python run for all OS	run-script-os		MIT	
102	CSS Framework + Components	CLR Icons	13.0.2	MIT + SIL Open Font License version 1.1	SIL Open Font License version 1.1 only for Fonts

Last Name	First Name	Value					
Sulzbach	**Lara**			**3.00**	**#DIV/0!**		
Oberndörfer	**Florian**	3					
Schnell	**Oliver**						
Klingenberg	**Christoph**						
Regl	**Amelie**			0	No size		
Engelhard	**Dirk**			1	Trivial size		
Rauen	**Moritz**			2	Small size		
Garbe	**Valentin**			3	Medium size		
Deli	**Deniz**	3		5	Large size		
				8	Very large size		
				13	Too large (size)		
How to play planning poker							
1. Everyone type their number into their value field, don't hit return yet							
2. Someone, perhaps a product owner, count down 3.. 2.. 1..							
3. Then, everyone hit return to submit their value							