



PRESENTATION: JUNE 2025

01

SARAH JANSSEN

Internship @Beja

•••

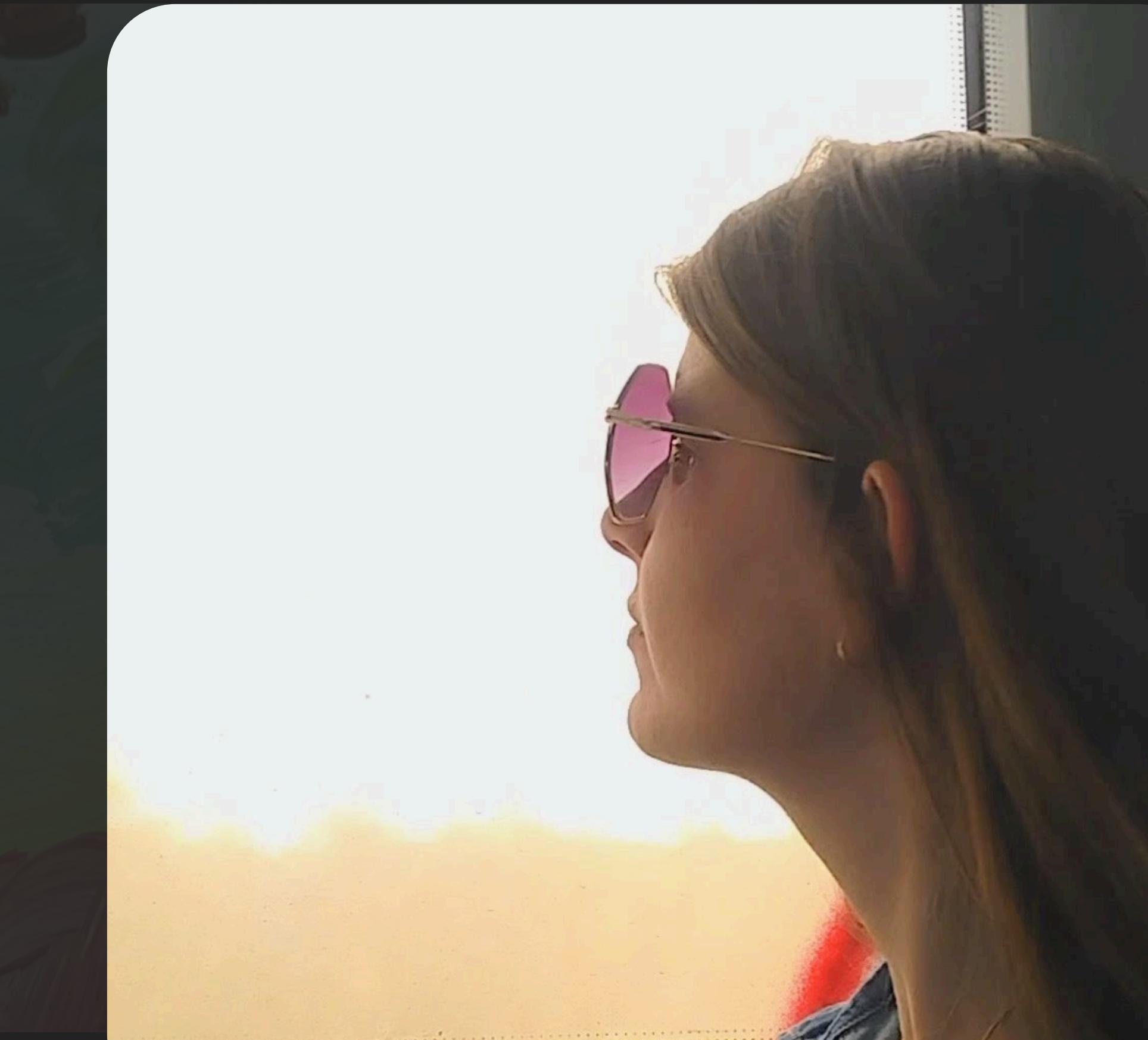
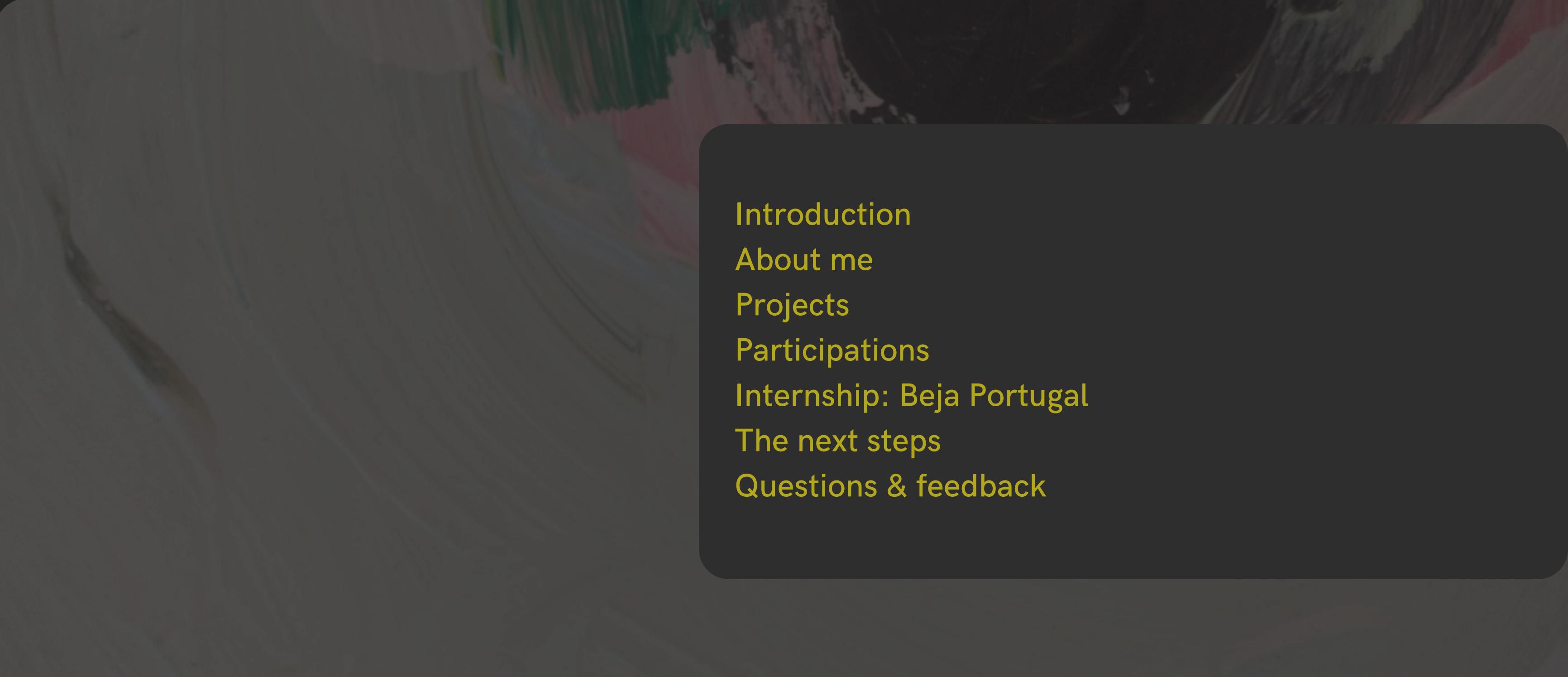




Table of Contents

02

- 
- Introduction
 - About me
 - Projects
 - Participations
 - Internship: Beja Portugal
 - The next steps
 - Questions & feedback



About me

SARAH JANSSEN

03

Sarah Janssen

Who am I?

Perfectionist

Eager to learn

Studies

Graphic design

Medical secretary

Why IT?

My partner sparked my dream



...



JUNE 2025

04

My skills

Soft skills



Problem-solving



Time management



Adaptability



Attention to detail



Creativity



Teamwork



Communication

Hard skills



C#



React



JavaScript



HTML, CSS



Python



Bootstrap

Tools



GitHub



WordPress



Docker



Jira



PROJECTS

05

My Projects

View more at
www.sarah-janssen.be



The image displays the Travel blog application interface, showing both a web browser view and a mobile phone view side-by-side.

Web View (Top): Shows a colorful collage of travel-related icons including hot air balloons, palm trees, the Statue of Liberty, the Eiffel Tower, and the Colosseum. Overlaid text reads "Discover destinations" with a green "click here" button. The top navigation bar includes "Home", "Destinations", "Blogs", "Profile", "Hi, test!", and "Log out".

Mobile View (Bottom Left): Shows the "Welcome to the Travel App" screen. It features a search bar with "Search blogs" and a "Login" button. Below it, there's a "Featured Destinations" section with "Oceania" and "South America" categories, each with a thumbnail image. Under "Latest Blogs", there are two entries: "Beaches of Bali" with a thumbnail of beach lights, and "Backpacking South America" with a thumbnail of a city skyline. At the bottom are three navigation icons: "Home", "Destinations", and "Blog".

Inset View (Bottom Right): Shows a detailed view of a blog post snippet. The visible text includes "adduco sustineo vestigium.", "ctor casus abundans.", "o cunabula contego.", and "us neque cupio.".

Travel blog

Key Features

- User Registration and Authentication
- Blog Management
- Commenting System by comments
- Mark favorites

Technologies

- Web application in Next.Js
- Mobile application in React Native
- Supabase for data service



06

More than just programming!



Meetups

AI, ethical hacking, Web 3.0, ...
Learn from experts
Connecting and networking

Hackathon

Hack The Future
Power apps
Offer travel to space
Me and my teammate: 4th place

Robocup and AI Show

Meet Lieven Scheire
Learn about robotica
AI show



07

How can I learn as much as possible?

By expanding to different platforms.

...



ERASMUS TRAINEESHIP IN BEJA

08

Erasmus in Portugal



I arrived as a stranger and came back with a (second) home.

...



ERASMUS TRAINEESHIP IN BEJA

09

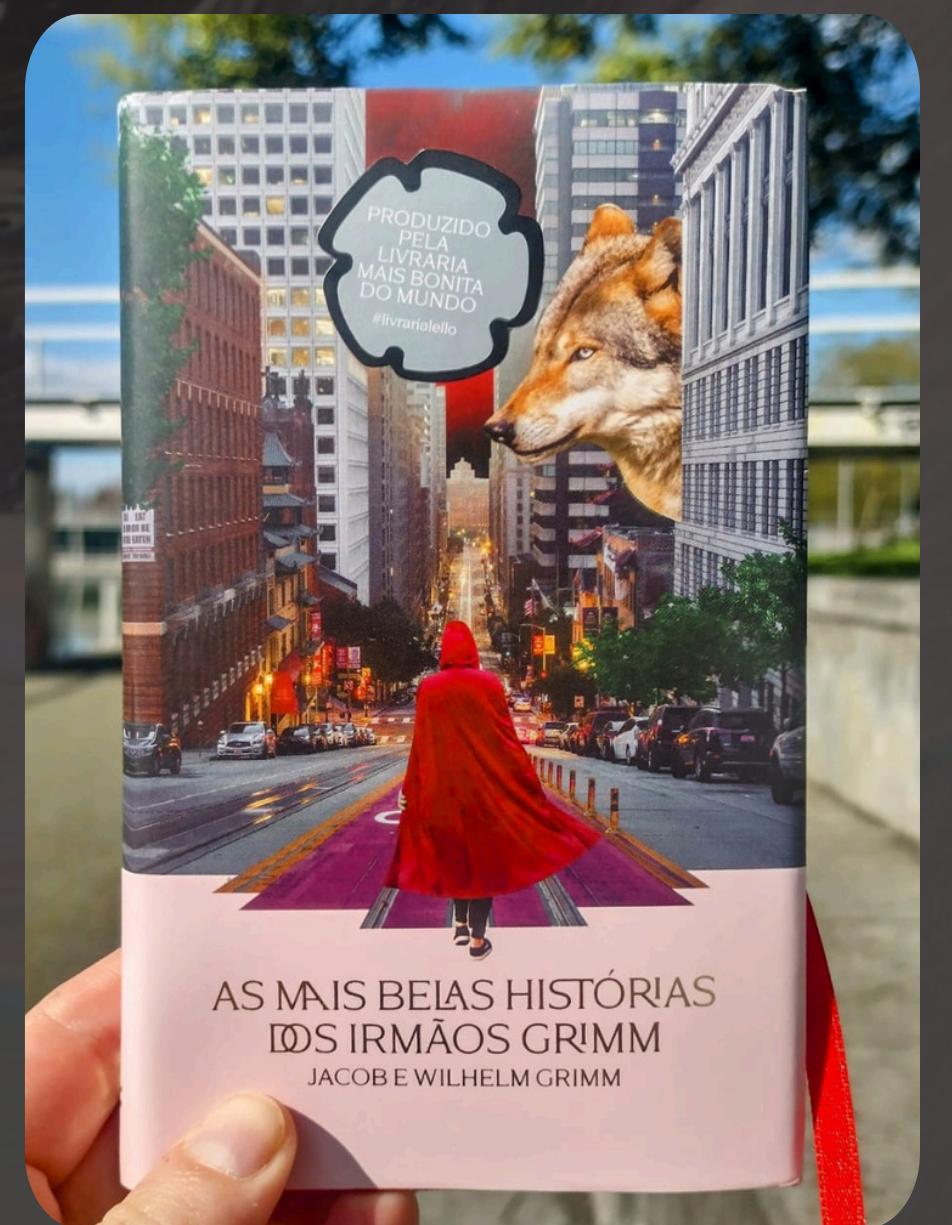
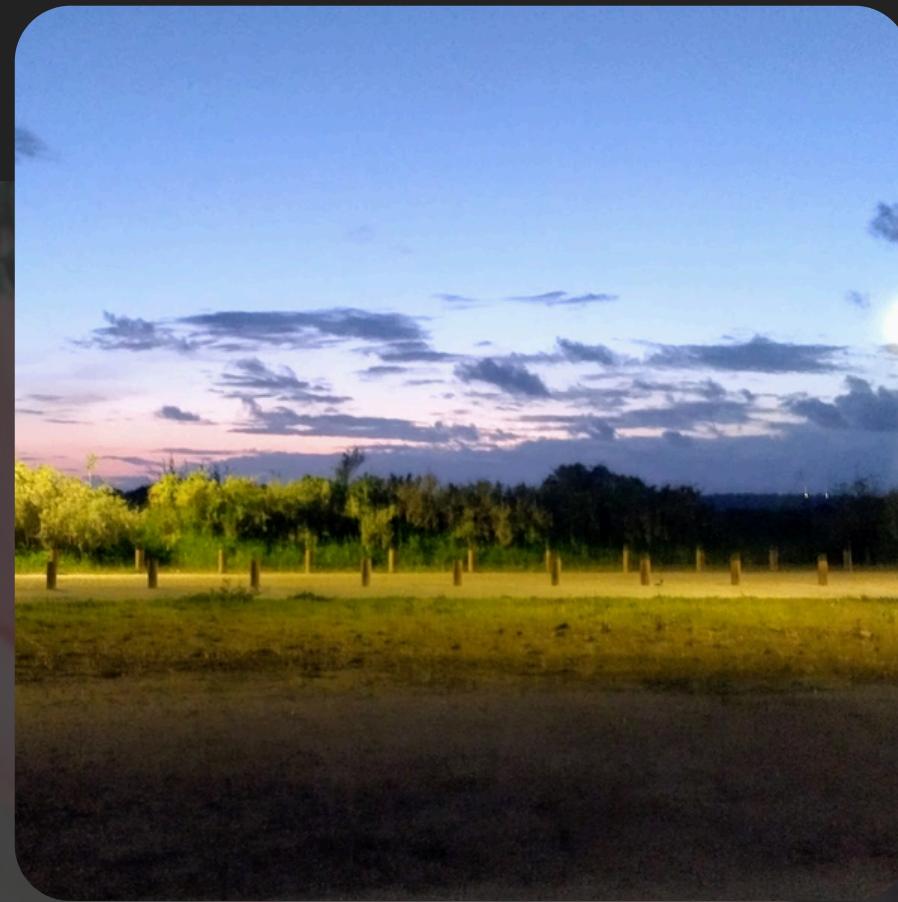
It was a great experience!

Erasmus in Beja

- Discover a new culture
- Learn about local products
- Meet new people
- Learn and improve languages

Growth

- As a developer
 - Learning new tools and technologies
- As a person
 - Independence and critical thinking





My Project

The screenshot shows a web application interface titled "Water Reports Application". At the top, there is a navigation bar with links: Overview, History, Current values, Configuration, Read Me, a search bar, and a "Search" button. Below the navigation bar is a section titled "Dashboard:".

The dashboard features six cards arranged in a 2x3 grid:

- Sensor Data**: View water sensor measurements. Includes a "View sensordata Details" button.
- Sensor data Graphs**: View sensor data historical. Includes a "View historical Data" button.
- Sensor data table**: View sensor data table. Includes a "View table" button.
- Documentation**: Go to the documentation section. Includes a "Read documentation" button.
- Dashboard Management**: Go to the grafana dashboard. Includes a "Manage Grafana" button.
- User Configuration**: Go to user configuration . Includes a "Go to Django Admin" button.

At the bottom of the dashboard, there is a copyright notice: "© 2025 Sarah Janssen. All rights reserved."

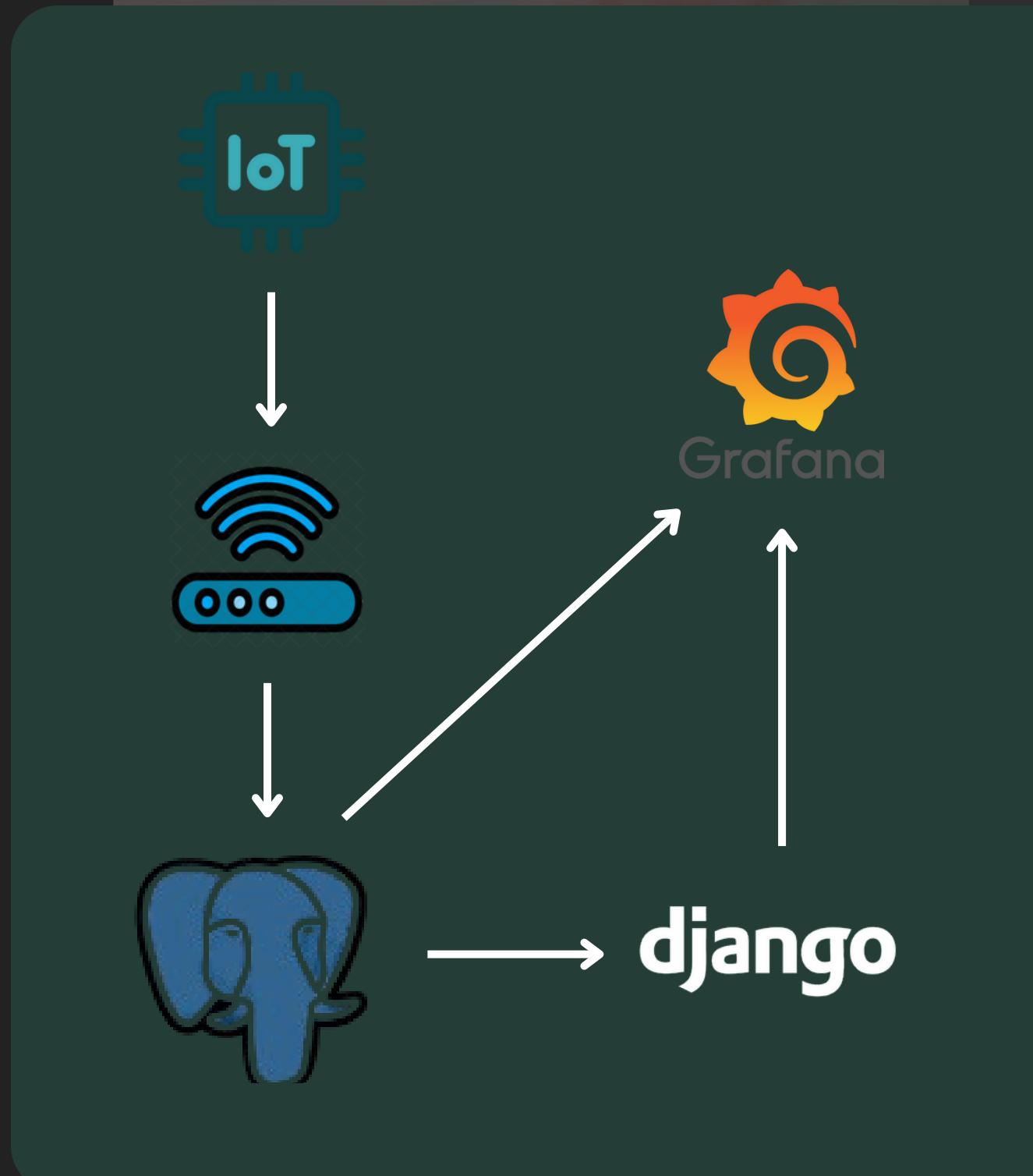
Water Reports Application

A web application for managing water measurement data.

Collects and stores water data in a structured database.

Displays measurement trends using interactive dashboards.





Context & Purpose

My role: make a reusable app by turning sensor data into insights

- Sensor data is collected by colleagues via an IoT platform
- Data is stored centrally in a shared PostgreSQL database
- My app reads and visualizes that data in a clear dashboard
- Users can search and filter measurements over time
- The app supports analysis through graphs and gauges

...

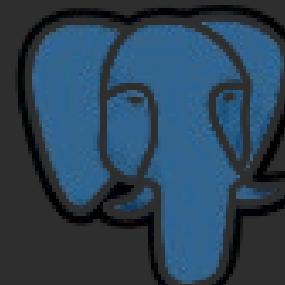


Application tools

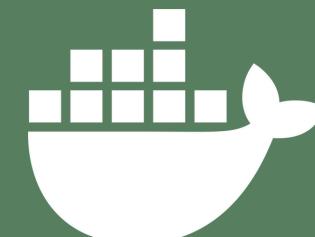
12



Programming language used for the back-end logic of the application.



Open-source relational database system.



Container platform to run the application consistently across environments.

django

Web framework based on Python.



Tool for data visualization and monitoring.

cypress

Front-end testing tool for automated browser tests.



Python

13

Core programming language used for backend logic

- Chosen for its simplicity, readability, and strong ecosystem
- Used in combination with Bootstrap to build a responsive and visually clean user interface
- Easy to learn, large community
- - Slower than compiled languages for intensive tasks

The image displays three screenshots of a Django-based web application interface:

- Dashboard sensor data:** Shows real-time sensor data for Temperature (15.3 °C), pH (7.15), and dissolved oxygen (6.60 mg/L) with corresponding gauges and "view summary" buttons.
- Historical Sensor Data:** Shows historical data for Temperature, pH, and Dissolved Oxygen over time from 13:15 to 13:30. Each section includes a line graph and a "View details" button.
- Project Read Me:** A sidebar menu with sections: Water Management System, Features, Project Structure, Prerequisites, and Installation.

...



14

Django

Full-stack web framework for rapid backend development

- Offers built-in tools (routing, ORM, admin) to accelerate development
- Powers routing, models, and admin features in your app
- Secure and structured
 - Less flexible than lightweight frameworks

The image displays a grid of 12 screenshots of the Django admin interface, arranged in a 3x4 grid. Each screenshot shows a different part of the admin site, such as user management, site administration, and specific data entry forms for water reservoirs. The interface is characterized by its dark blue header and light blue sidebar, with white text and standard UI elements like buttons and dropdowns.

- Row 1: User management (list view), Site administration (list view), Water reservoir management (list view), Water data entry form.
- Row 2: User management (change view), Cabinet management (list view), Water data entry form, Water data entry form.
- Row 3: User management (change view), Cabinet management (change view), Water data entry form, Water data entry form.



PostgreSQL

The screenshot shows the pgAdmin 4 interface. At the top, there's a navigation bar with tabs for 'readme.md', 'docker-compose.yaml', 'console', and 'waterdatatest2'. Below the navigation bar is a toolbar with various icons for database management. The main area displays a tree view of database objects under 'dummy_postgres@localhost'. Under the 'public' schema, there's a 'tables' node containing 'waterdatatest2'. This table is shown in a grid view with the following data:

	timestamp	temperature	ph	dissolved_oxygen	conductivity	redox_oxygen
1	1712060000	15.123	7.100	6.20	450.00	12.00
2	1712060600	14.987	7.300	6.50	455.20	11.90
3	1712061200	15.500	7.250	6.40	452.10	12.10

Reliable and advanced database for structured data

- Chosen for its strong query support and open-source nature
- Stores water measurements and connects to Grafana
- Powerful and scalable
 - More complex than SQLite



16

Docker

```
grafana
grafana/grafana:lates 3000:3000
2025-05-08 21:34:51 dummy_postgres | 2025-05-08 20:34:51.795 UTC [27] LOG: checkpoint starting: time
2025-05-08 21:34:51 dummy_postgres | 2025-05-08 20:34:51.822 UTC [27] LOG: checkpoint complete: wrote 3 buffers (0.0%); 0 WAL file(s) added, 0 removed, 0 recycled; write=0.004 s, sync=0.002 s, total=0.028 s; sync files=2, longest=0.001 s, average=0.001 s; distance=0 kB, estimate=0 kB
2025-05-08 21:39:53 grafana logger=cleanup t=2025-05-08T20:39:53.860105867Z level=info msg="Completed cleanup jobs" duration=34.534182ms
2025-05-08 21:39:54 grafana logger=plugins.update.checker t=2025-05-08T20:39:54.316145739Z level=info msg="Update check succeeded" duration=132.188264ms
2025-05-08 21:49:53 grafana logger=cleanup t=2025-05-08T20:49:53.826006742Z level=info msg="Completed cleanup jobs" dura
```

Container platform to deploy and run your app anywhere

- Chosen to ensure consistent environments across machines
- Packages Django, PostgreSQL, and other services
- Easy deployment and portability
 - Initial learning curve



17

Grafana

Visualization tool to monitor and explore water data

- Chosen for real-time dashboards without custom coding
- Displays measurement trends using time series and gauges
- Beautiful visuals, easy setup
- – Read-only, not a reporting tool





The collage includes:

- A large screenshot of a dashboard titled "Testing the project" showing sensor data for Temperature (15.3 °C) and pH value (7.15 pH).
- A screenshot of the Cypress Test Results interface showing a green progress bar and 2 specs passed.
- A screenshot of the Cypress Cloud interface showing a dissolved oxygen reading of 6.60 mg/L from a Dissolved Oxygen Sensor.
- A screenshot of the Cypress Cloud interface showing a dashboard with sensor data for Temperature Sensor, PH Sensor, and Dissolved Oxygen Sensor.
- A text overlay at the bottom of the collage stating "Cypress runs & push the results in the Cypress Cloud." with a large red "APPROVED" stamp.

Cypress

Frontend test automation tool

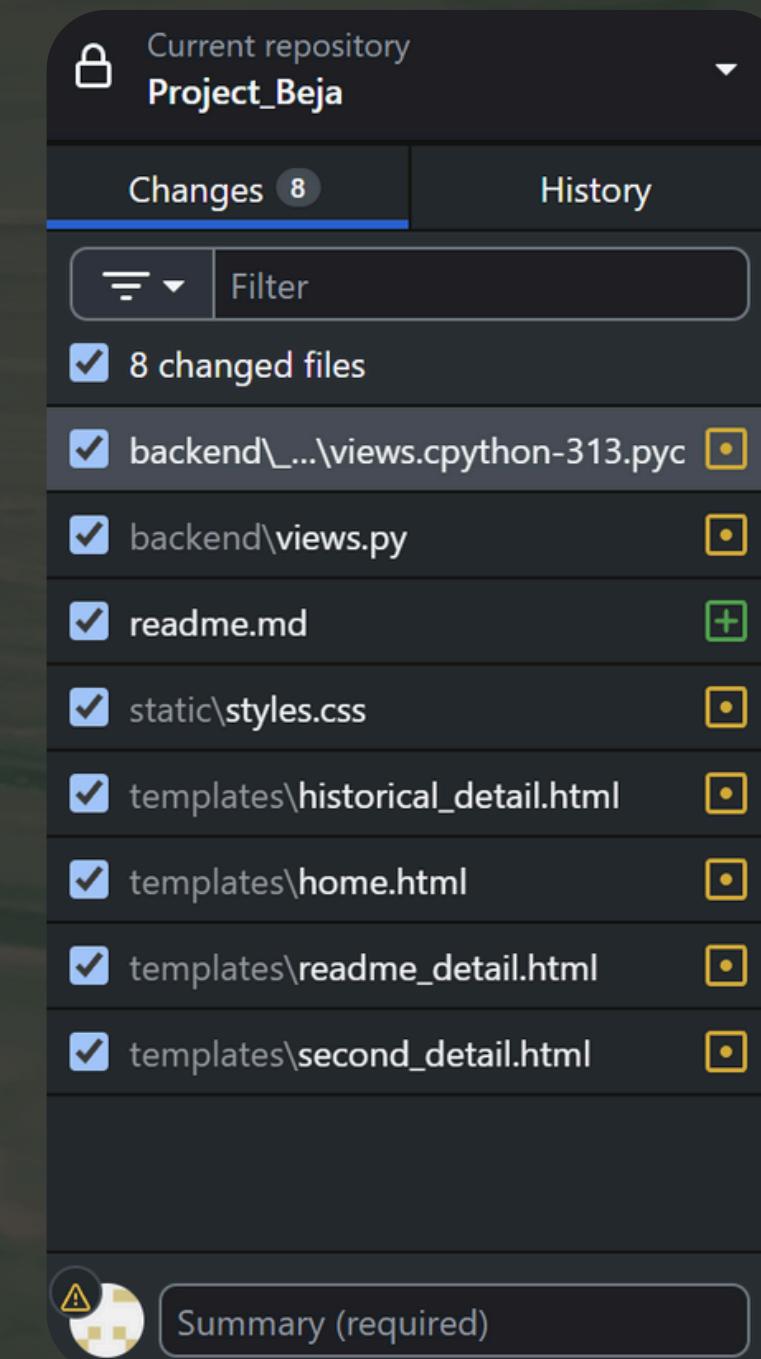
- Chosen to automatically test UI components like the search bar
- Verifies that user actions and interface behave correctly
- Fast feedback in browser
- - Always in users perspective, application needs to run before test



Github desktop

Visual Git tool for version control

- Chosen to simplify Git usage without using command-line tools
- Used to track changes, manage branches, and push code to GitHub during development
- Beginner-friendly, clear commit history
- - Less flexible than command-line Git for advanced tasks

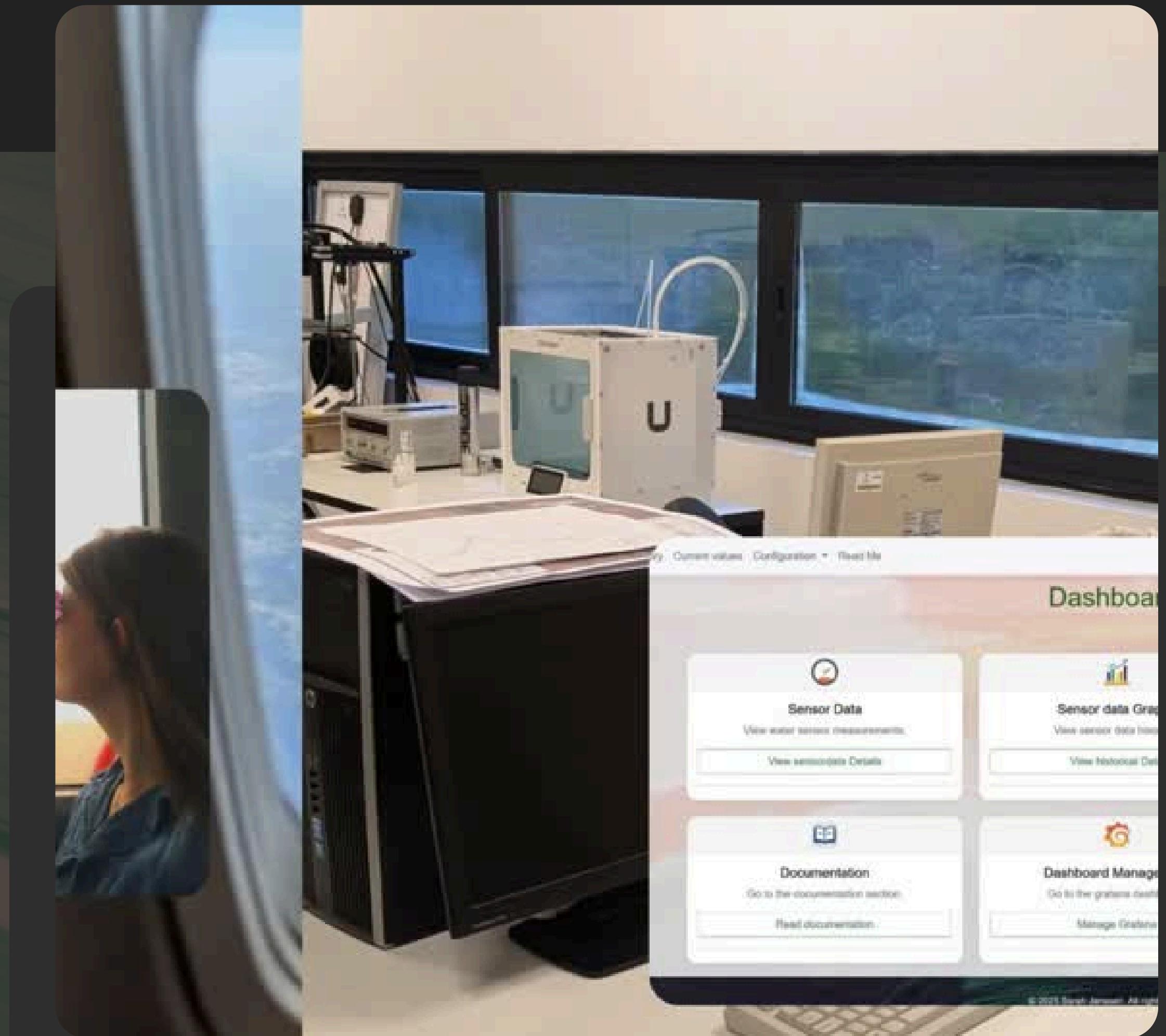




PROJECT BEJA

System demo

20





21

In Summary

Who Am I as an IT Professional?

- Frontend Focus
- Creative, structured, eager to learn
- React, Tailwind, TypeScript
- Customer-oriented - background in healthcare and retail
- International experience (Beja - Erasmus)



...



WHAT'S NEXT?

JUNE 2025

The Next Steps

22

June

Aclimatize
Stabilize

July/August

Search for my
dream job
Finalise projects

September

Working on a
great start

October

Back to Beja to
network
Build the future...

For the coming months

...



JUNE 2025

23

Thanks for your attention.

It is now the time for questions &
feedback.

LinkedIn



Email Address

janssen.sarah1991@gmail.com

Website



www.sarah-janssen.be

•••