LE GOUVERNEMENT DU GRAND-DUCHÉ DE LUXEMBOURG Ministère de l'Éducation nationale, de l'Enfance et de la Jeunesse		**DT Computer Technician Final Integrated Project**
Education:	DT Computer Science	**Work assignment**
Modules:	REPIF1, REPIF2, PROFI	
School	2025/2026	

Work assignment

Table of contents

1 Topic

You work as a technician for a company that specializes in manufacturing weather stations and indoor climate sensors. In this context, your employer would like to expand its product range with a new product that enables customers with little technical knowledge to monitor the indoor climate in different rooms of their home and share the collected data with other users.

The *Portable Indoor Feedback* Station is to consist of a wooden housing with a display for showing the data. To make it as easy as possible for customers to use, it should only be necessary to connect the device to the power supply and the home network.

In addition to the station, an online platform is also to be created where customers can register their purchased stations, view their data, and share it with other users.

As a specialist in the technical aspects of such a project, you are now tasked with creating a prototype of the planned product.

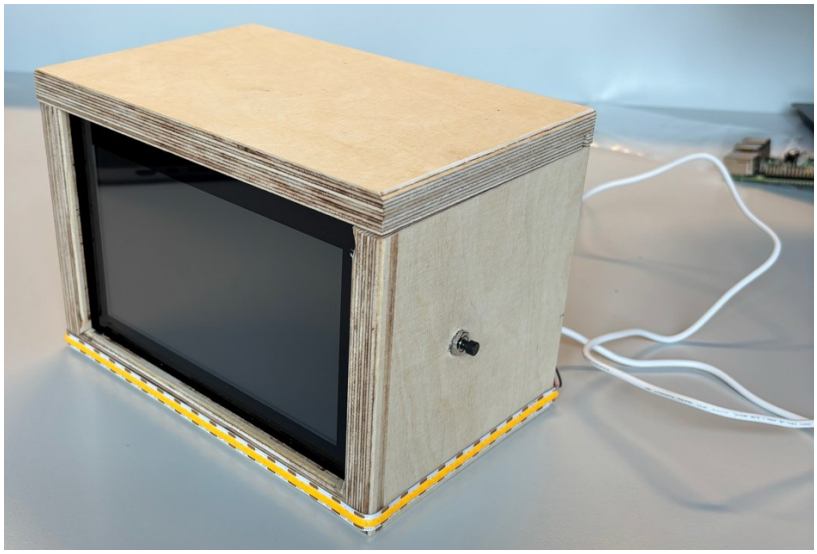
2 General project description

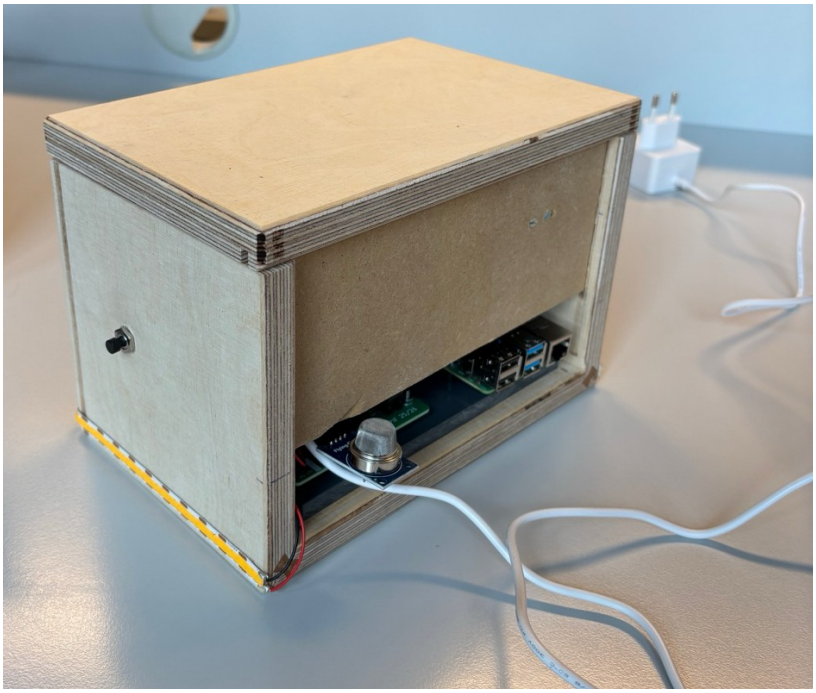
The system to be created consists of the following components:

2.1 Station

The planned system consists of stations comprising a wooden housing with a built-in display for showing the data. The station is based on a Raspberry Pi computer to which various sensors for measuring the indoor climate are connected via a circuit board.

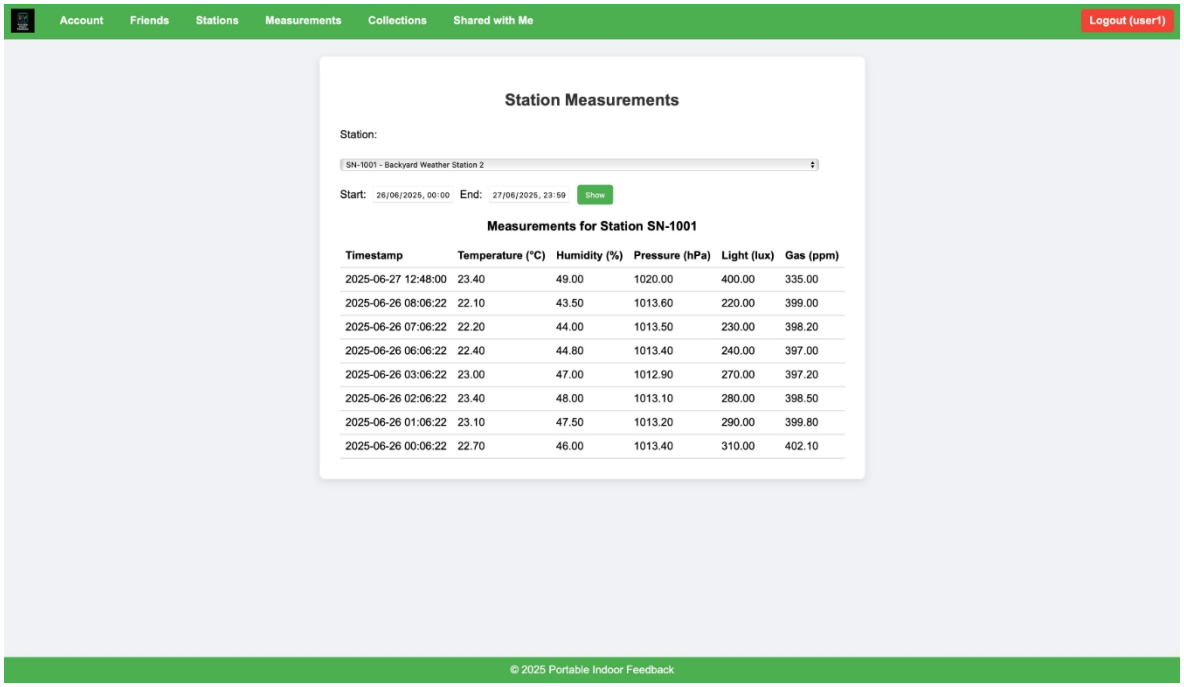
The station is connected to the Internet via Ethernet so that the measured data can be transmitted to the company's central server.





2.2 Server and website

The company's central server is used to collect the measured data from the various stations and store it in a database. The server also hosts the website, which allows customers to view and share their data. To prevent data loss and protect against breakdowns and cyberattacks, regular backups of the database and, ideally, the entire server should be made.



3 Calendar

KW	from	to	Lab	Studio	Presentations
Internship					
44	10/27/25	02/11/25	**Read introduction and project assignment / Preparation**		
Vacation					
46	10/11/25	11/16	**Raspberry Pi**	**PVC mount**	
47	11/17/25	11/23/25			
48	11/24/25	11/30/25	**Database**		
49	01/12/25	07/12/25			
50	08/12/25	12/14/25	**Website**	**Printed circuit board**	
51	12/15/25	12/21/25			
Holidays					
02	01/05/26	11/01/26	**(Planning)**	**Power supply**	
03	12/01/26	01/18/26			
04	01/19/26	01/25/26		**Sensors 1**	
05	01/20/26	02/01/26			**Interim presentation REPIF1**
06	02/02/26	08/02/26	**Web database server**	**Sensors 2**	
Holidays					
09	02/23	01/03/26	**(Planning)**	**Sensors 2**	
10	03/02/26	03/08/26		**Assembly**	
11	03/09/26	03/15/26	**Data transfer**		
12	03/16/26	03/22/26		**Lighting**	
13	03/23/26	03/29/26	**Backup server**		
Vacation					
16	04/13/26	04/19/26	**Backup server**	**Test**	**Interim presentation REPIF2**
17	04/20	04/26	**Final sprint**		
18	04/27	05/03/26			
19	04/05/26	May 10			
20	May 11, 2026	05/17/26			
Preparation of the defense					
23	06/01/26	06/07/26	/	/	**Defense**

4 User Stories Workshop

4.1 PVC holder

The first step in manufacturing the station is to create an internal framework consisting of a PVC plate to hold the touch display and a second PVC plate to hold a circuit board and the Raspberry PI.

Both sheets are screwed together to form a basic frame. This makes it possible to transport the display, the Raspberry PI, and the circuit board without damaging the components. Later, the completely assembled basic frame is then inserted into a wooden housing to present a clean and finished product.

4.2 Circuit board

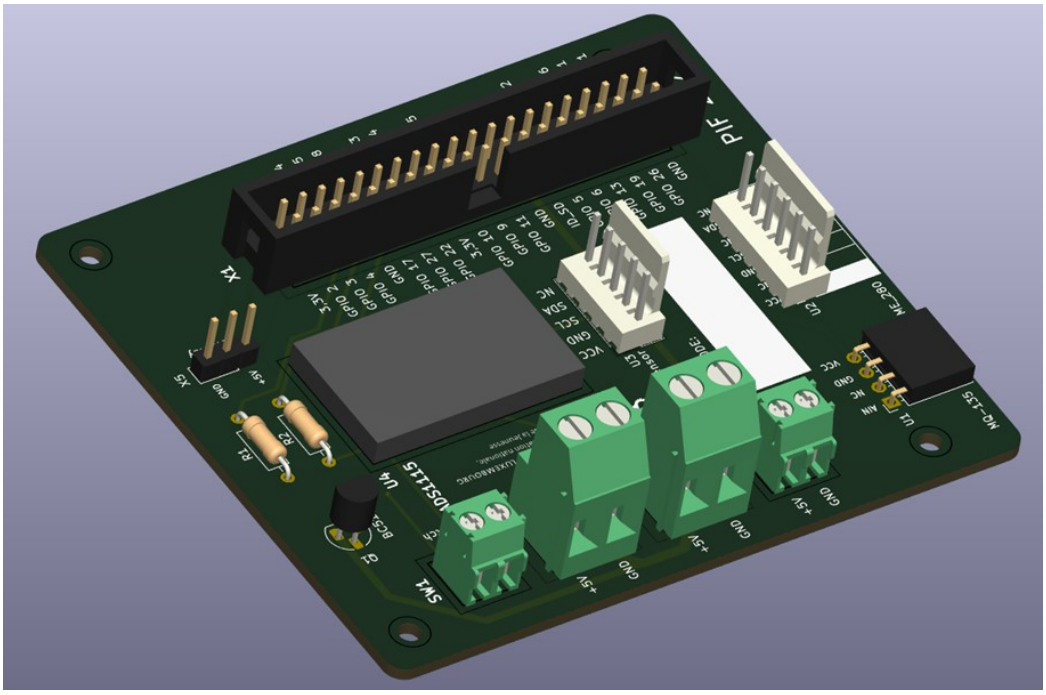
In the second step, a circuit board is manufactured to serve as an interface between the Raspberry PI, the sensors, and the actuators (lighting).

The board has a 40-pin connector, which is connected to the GPIO ports of the Raspberry Pi using a 40-pin flat connector. Sockets are provided for connecting sensors so that they can be plugged in and easily replaced if problems arise.

There are 4 screw terminals on the circuit board – 2 for the power supply, one for connecting an external button and one for connecting the LED strip.

A 3-pin connector is used to connect the power supply for the touch display. Finally, there is a

small driver circuit for switching the LED on the circuit board.



4.3 Power supply

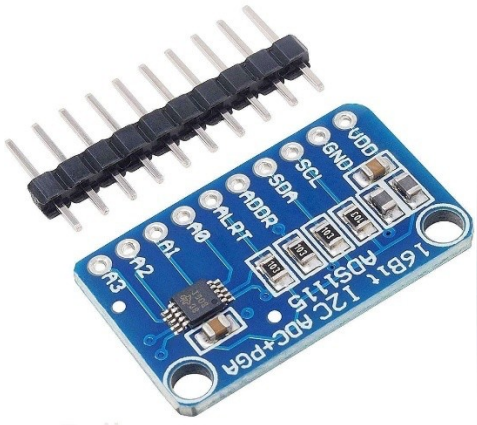
The station is powered by the external power supply unit supplied with the Raspberry Pi. This provides the necessary +5V DC voltage, is short-circuit-proof, and has sufficient power reserves.

To prevent a voltage drop at the Raspberry, the power supply must be disconnected so that the load (LED) and the Raspberry Pi run in parallel on the power supply. This ensures a steady power supply to the Raspberry Pi.

4.4 1 sensors

The ADS1115 16-bit converter is required to process the analog output signal of the MQ-135 gas sensor module into a digital output signal. This is then sent to the Raspberry via the I2C interface so that it can be further processed.

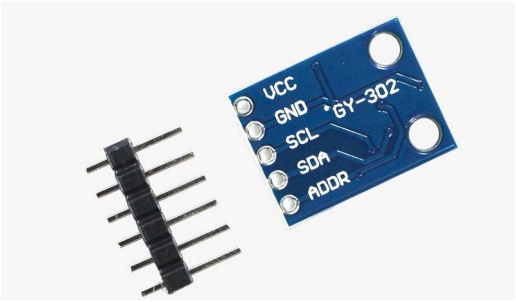
The MQ-135 gas sensor module can determine air quality by detecting air particles. The measuring range is 200-1000 ppm (parts per million). For this prototype, the concentration of CO2 is to be measured.



4.5. e 2 sensors

The BME 280 barometric sensor module can detect temperature, air pressure, and humidity. These are to be sent as a digital output signal via the I2C interface to the Raspberry Pi for further processing.

The light sensor module can measure light intensity with an accuracy of 16 bits. The light intensity to be measured is 1-65535 lux. The data is also forwarded via the I2C interface.



4.6. Assembly

Once the sensors have been put into operation and checked for correct functioning, assembly can begin.

The wooden housing consists of three parts: a frame, a removable cover, and a rear panel that slides into place. The power supply is connected. The gas and environmental sensors are plugged into the designated circuit board, and the light sensor is mounted on the rear panel. The device is put into operation and checked to ensure it is functioning properly.

4.7. Lighting (LED)

Finally, outdoor lighting (LED) is installed for decorative purposes (can also be used as a night light). By default, the lighting should turn on automatically when the measured light falls below a certain level, i.e., when it gets too dark. However, it is possible to consider more complex strategies here or to use the lighting as an alarm, for example, if a measured value reaches a critical range. Optionally, a button can also be installed to manually switch the lighting on and off.

5 User Stories Labo

5.0 Preparation

In the first week of the project, your teachers will provide you with the necessary explanations about the project process. You should familiarize yourself with this process and the documents you have been given and read everything in detail.

Then you should select a planning tool to use for planning your project and the user stories to be planned. Use this tool to create the basic framework for your planning and think carefully about how you want to work with the tool during the course of the project.

5.1 Raspberry Pi

Each station is controlled by a Raspberry Pi 4. This is responsible for reading the sensors at regular intervals and displaying the collected data locally. It also ensures that the data is sent to the company's central server so that it can be accessed from the website.

The Raspberry Pi runs the 64-bit version of the Raspberry Pi OS. All built-in sensors are addressed via the I²C interface. Data is read and sent using a Python3 script (firmware), which can be configured via an external file (`config.toml`). This script detects whether sensors are connected. If this is not the case, it falls into test mode and always delivers the same test data, with only the time changing. Details on usage can be found in the help (`-h/--help`). In addition to sending the data to the server, the script also writes all data to the standard output.

5.1.1 How the firmware works

```
001-007     General imports.
009-042     Definition of command line options and help.
044-048     Sensor-specific imports.
051-052     Start of the script.
053-054     Parsing the command line.
056-058     Setting up the GPIO pin for lighting control.
060-062     Reading the configuration file.
064-091     This part is only executed if the sensors are to be read out
            .
065-068     Check whether all I2C addresses are configured; if not
            the case, an error message is issued and the system switches to
            mode without sensors.
070-091     Creation of the I2C bus and sensors. If an error occurs here
            , it is displayed and the script switches to the
            without sensors.
093-133     This is the main loop.
095-130     If an error occurs when reading the sensors, it is
            and a new loop cycle is started.
097-102     The sensors are read out.
103-108     If the script runs without sensors, fixed values are used.
110-110     Saving the current time.
112-116     Switch lighting on/off based on the current and
            configured brightness.
118-123     Outputting the values in the standard output, with or without
            units.
```

```
125-127     Call the function to send the data.
132-133     The configured timeout until a new measurement is taken.
137-152     Function for sending the data.
139-147     Create the object to be sent, consisting of the serial number, timestamp, and measured values.
148-152     Attempt to send the data to the configured script. If the connection is not successful within one second, an error message is written to the standard error output and the script continues.
155-156     Call the main function.
```

5.1.2 Configuration file

The configuration file is created by the technician when configuring the station and contains the serial number and measurement interval as well as the I2C addresses of the sensors and information about the company's main server. For this prototype, you can choose the serial number yourself. However, make sure to use the same serial number in the configuration file and later in the database, as this is used to identify the individual stations when transferring data to the server. For this purpose, you will receive the following skeleton of a configuration file. You must uncomment the individual parameters and configure them with appropriate values.

```
#station_serial =                  # Serial number of the station

interval = 2                       # Time between measurements

toggle_intensity = 100             # The light level at which the LED should be turned on

[i2c]                              # I2C-Addresses of the sensors
#bme280 =
#bh1750 =
#mq135 =

[destination]                      # Information about the server
#ip =                             # IP-Address or Hostname of the Server
#path =                           # Path to the script that accepts the measurements
```

The configuration file must be located in the `/var/station` folder!

The I2C addresses can only be configured after the sensors have been connected. Until then, the script runs in test mode. Once the addresses have been configured and the sensors connected, the script generates correct data.

The IP address can be configured directly, as can the path of the PHP script for data transfer (on the web server), but the `--offline` option should be used until this PHP script has been created.

This file is not edited by the customer.

5.1.3 Local display

To display the data locally, a web server runs on the station using Docker, which hosts a website with the current data. This page is displayed in kiosk mode on the station's display. Communication between the script and the container takes place via

a shared folder. The design of this page is kept very simple and should be personalized and made more appealing if possible.

Time	2025-07-15 16:03:34
Temperature	24.91 °C
Humidity	38.43 %
Pressure	978.09 hPa
Light	454.17 lux
Gas	2,628.00 ppm

5.2 Database

In order to store the measured data from the individual stations properly, you should design a model for a database. This database will later be used to display all data correctly on the website, thus giving customers the opportunity to manage their stations and data.

In the first phase, you should design a conceptual data model (MCD/CDM), which you should then convert into a logical data model (MLD/LDM). The resulting database should be created in a local environment and ideally filled with meaningful sample data. LLMs and similar AI tools can be helpful here in obtaining realistic data. This data will later be used to test and use the website even when no connection to stations is guaranteed. When modeling, make sure to use correct nomenclature and meaningful names as well as logically selected data types and parameters.

Your database should be able to store the following types of data:

Users	Users are identified by a unique username. Each user must also provide their full name and an email address. There are both normal users and administrators with special rights.
Stations	In addition to the serial number assigned by the company (e.g., SN-1001, SN-3102, ...), the station can be assigned a name and a detailed description can be entered.
Measurements	A measurement contains a timestamp (date and time) and the measured data (temperature in °C, humidity in %, air pressure in hPa, light intensity in lux, air quality in ppm).
Collections	Measurements can be grouped into collections. For this purpose, each collection can be assigned a name and a detailed text description.

A station is created in the database by an administrator user as soon as the station has been produced. If a customer has purchased it in a store, they can register it after creating their user account. A customer can own multiple stations, but a station belongs to only one user.

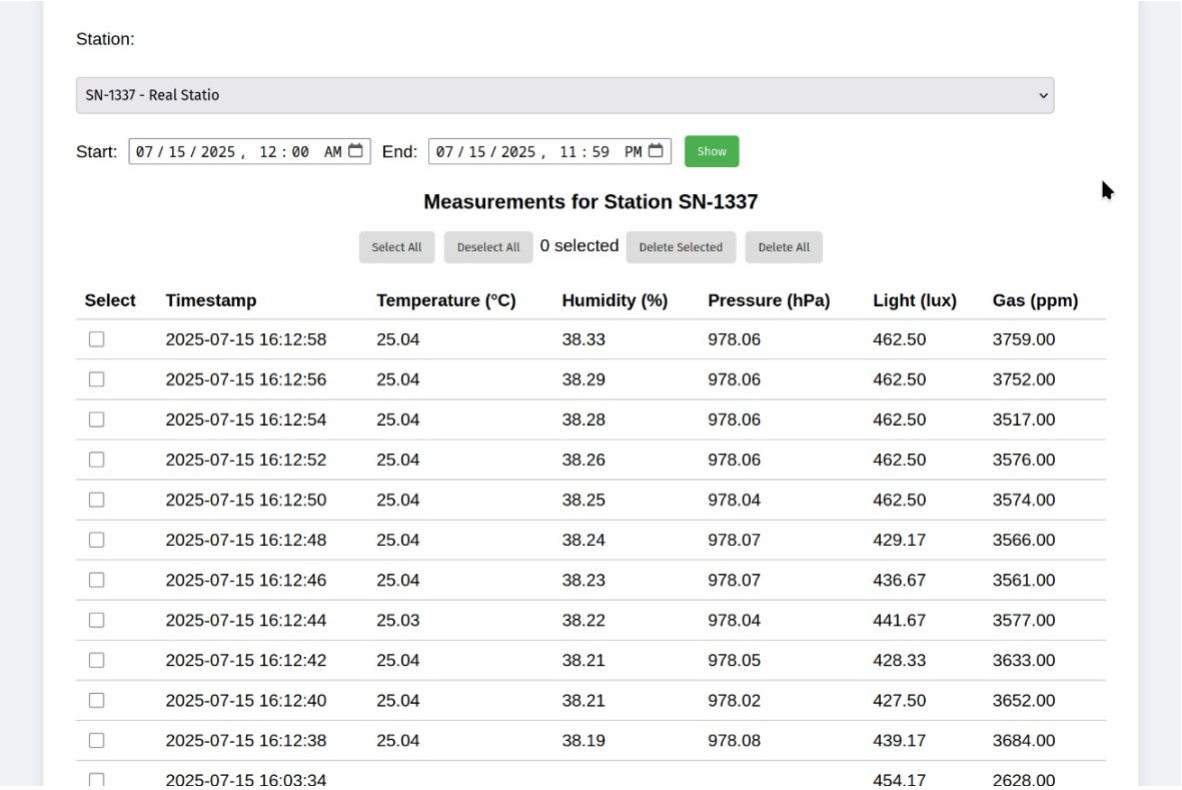
Measurements are recorded by stations, with each measurement being assigned to a single station.

Collections contain measurements, and empty collections are also allowed. The same measurement can also be included in multiple collections. A collection is created by exactly one user, and the same user can create as many collections as they want.

Users can add each other to a friends list. Of course, no one is obligated to have friends, and it is possible to have multiple friends at the same time. A user can give other users access to their collections by sharing the collection with them. It is possible to have access to multiple collections, and the same collection can also be shared with multiple users at the same time without any problems.

5.3 Website

The website is the heart of the project, as it allows users to view the measurement data from their stations and share it with other users. To this end, you should create a website that is as user-friendly and appealing as possible, using the database you have created. You should use the techniques and languages you have learned, such as HTML, CSS, JavaScript, and PHP. You are also allowed to integrate and use external libraries such as Bootstrap or ChartJS. Comment your source code to make it as clear and understandable as possible for other employees and for later extensions/maintenance of the website. The finished website should be uploaded to the server you have installed and use the database created on this server.



User accounts

First, it should be possible to create a user account. After creating the account, the user does not have administrator rights, but can log in to the website using a username and password. They now have access to the normal user area.

User area

After a user logs in, they see a welcome page. This should initially be kept very simple and only contain a welcome message. In a later phase, it will be expanded to include a dashboard that displays the latest measurements from a station selected by the user, ideally using charts or similar graphical elements. Users can also edit their user accounts and change their data.

In this area, it is possible to register stations after a customer has purchased them in the store. In this simplified version of the website, this is to be done using the serial number, which the customer enters directly. Later, this will be simplified by allowing the user to scan a QR code provided, for example. The customer can then assign names and descriptions to their stations to make it easier to keep track of them.

The user can now view the measurement data recorded by their stations. To do this, they can select one of their stations and a start and end date and time. The corresponding measurement data is then displayed. This can be a simple tabular view, but later, if possible, it will be expanded to include graphical views such as diagrams.

It is also possible to create collections of measurements. To do this, the user again selects one of their stations, a start and end date and time, and gives the collection a name. The measurements recorded during this period are now automatically combined into a collection. It should also be possible to rename or delete collections that have already been created and to obtain a list of the measurements they contain.

Collections can be shared with users who are friends with the creator of the collection. Each user also has the option of viewing the collections shared with them, but without being able to edit them.

To this end, it should be possible to add other users as friends. In the initial phase, this should be done simply by entering the user name. When a user name is entered, a friendship between the two users is automatically established. Later on, it would be better to implement a system based on friend requests to give users more control over their privacy. A user can view their friends list and remove other users from this list. In this case, it would make sense if collections shared between users were no longer shared automatically.

Administrator area

Administrators are special users employed by your company who monitor and manage the system. In addition to everything that normal users can do, they should have the following functionalities available on the website.

Administrators should be able to manage all user accounts. This includes editing all data and deleting user accounts. New user accounts can also be created directly in the administrator area. Administrators can promote other users to administrators or revoke their administrator rights.

In order for users to register their stations, these must first be created by an administrator in the administrator area. In addition, all stations can be edited (with the exception of the specified serial number). An administrator can also delete stations and change the owner of a station.

A large proportion of the operations that normal users can only perform on their own stations or collections can be performed by administrators on all stations or collections.

5.4 Web Database Server

All data is stored on a database server. This can be a virtual or physical machine. The same server also serves as a web server.

The operating system to be installed is the latest LTS version of Ubuntu Server. In addition, the Apache, MariaDB, PHP Stack, and OpenSSH packages should be installed and configured.

The host name and static IP address should be configured according to the instructions. The time zone, date, and time must be set correctly.

A database user (candidate's IAM code) must be created with the "All PRIVILEGES" privilege. It is not recommended to use *mysql_secure_installation*. You should use the *unix_socket* authentication method.

When configuring PHP, you should ensure that any PHP error messages can be displayed correctly later on. The PHPMyAdmin package should also be installed on the server and configured correctly.

All work steps for this stage must be clearly listed individually in your planning tool. Particular care must be taken when assessing the individual stages to be completed. When planning, it is also important to ensure that all stages can be realistically completed within the time allocated. In the event of unforeseen events or problems, the planning must be adjusted at any time on the timeline.

5.5 Data transfer

In order for the station you have installed to be able to transfer data to the server you have installed and the database created on it, the connection and data transfer must be ensured.

To this end, you should write a PHP script to be hosted on the server. This script should initially enable data to be transferred to the server. In a later, extended version, it should also be possible to query measurement data. The script should be accessible with the following POST parameters:

station_serial	Serial number of the station
timestamp	Date and time of the measurement
temperature	Temperature in °C
humidity	Humidity in %
pressure	Air pressure in hPal
Light	Light intensity in lux
gas	Particles in ppm

The data obtained from the script is then checked and inserted into the database.

Comment your source code so that it is clear and understandable and as easy as possible to expand and maintain.

For testing purposes, the script can be extended with a form that allows measurement data to be entered manually. This is helpful, for example, if a connection to a station is currently not possible but new data is needed for testing.

5.6 Backup server

A backup server is required to create regular backups of the data from the database server. For this purpose, a virtual machine with the current LTS version of Ubuntu Server is installed. During installation, the host name is set to include the student's IAM code. The static IP address assigned by the teacher is then configured and the network adapter of the VM is adjusted accordingly. After that, the operating system is updated, the time zone is set to "Europe/Berlin," and the OpenSSH server is installed to enable remote access.

A separate directory for database backups is created on the backup server, e.g. /srv/backup. The access rights must be set so that only the designated user has access. In addition, MySQL client software is installed to enable database connections.

For the daily backup, a shell script is created that backs up the project database from the web and database server and stores it in the backup directory. This script is given execution rights and is configured to run automatically every day at 10:00 a.m.

You can also set up an FTP server for external storage. You should create your own FTP user and set their home directory to the backup directory

. The configuration must be adjusted so that the user only has read access to the directory and cannot view any higher-level paths.

External connections are permitted on the web and database server; the configuration must be adjusted correctly for this. A special MySQL user should be created who is only allowed to access the project database from the backup server. The user is granted all rights to the project database.

6 Material list

Component	Quantity
2-layer printed circuit board 75 x 85 mm	1
40-pin straight connector strip R WS40G	1
GPIO ribbon cable for Raspberry Pi Model B+ (40 pins)	1
Metal resistor 0207 0.6W 10K	1
Metal resistor 0207 0.6W 56K	1
BC517 transistor	1
PCB connection terminal 2-pin RM 5	2
PCB terminal block, 2-pin RM 3.5	2
1-row angled socket strip 1 x 4 -RM2.54mm h=8mm	1
1-row angled female connector strip 1 x 7 -RM2.54mm h=8mm	1
Pin strip, 1 row, straight, 1 x 3 - RM2.54 mm	1
PCB connector, straight pin header with female connector (set with cable)	1
M3 x 12 countersunk head	3
M2.5 x 20 countersunk head	8
M2.5 x 10 cylinder head	4
M2.5 washers	8
M2.5 nuts	8
5 mm plastic spacer roller	8
Gas sensor module MQ-135	1
ADS1115 ADC module I2C	1
GY-BME 280 Barometric Sensor Module	1
GY302 / BH1750 Light Sensor Module	1
Raspberry PI 4B 2GB	1
Raspberry PI AC Adapter	1
Touch display PI Touch 2	1
Heat shrink tubing	
LED strip	
Switching cable 0.5 mm² black	
Switching cable 0.5 mm2 red	
SD card	1