

## Recruitment tasks - .Net Core & React.js

When applying for the position of fullstack developer, I was given recruitment tasks to complete:

1. Write an application in C# to retrieve information about entrepreneurs by TIN number. Once retrieved, display the information on the page and save it to the database. Use the WL Registry API (mf.gov.pl).

\* The use of React.js in the front-end will be an added advantage.

2. Choose one task from the following:

a. Write code in C# that:

- i. move the subtitles from the .SRT file 5 seconds and 880 milliseconds (00:00:05,880) forward,
- ii. will cut issues from the file that start at equal seconds after the shift and paste them into a new file,
- iii. will give new subtitle numbering for both files.

You can download the .SRT file here: [movie subtitles.srt](#).

- b. Write a simple application in React.js that allows you to measure the time of tasks performed. The app must allow you to start, name and finish a task (start - stop system). Let the completed tasks appear in a list that shows the names of the tasks and their duration. The application is not required to communicate with the server, it can only run on RAM.

I completed tasks 1, 2a and 2b within the deadline of one week. Since I had previously used Vue.js and Angular.js, and was not familiar with React.js - I had to learn it very quickly. Source codes of my work are available at <https://github.com/zbyszex1/ANTHEAP>.



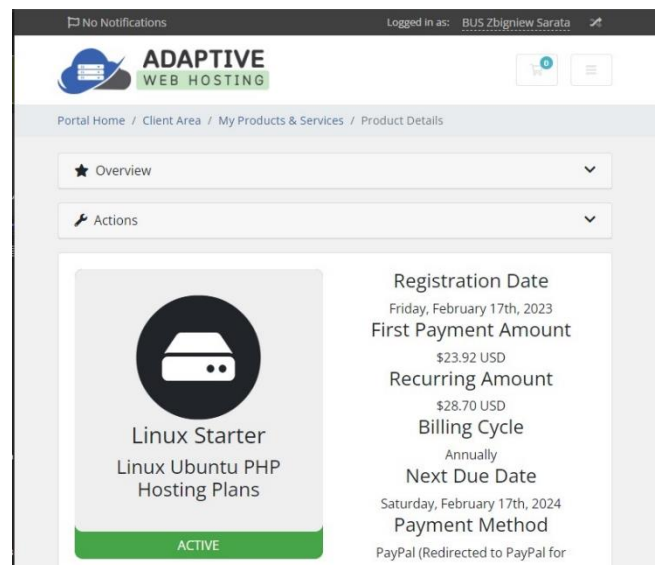
## Indexing - .Net Core 6 web application hosted on Linux server

Hosting .Net Core web applications in an Azure environment seems natural and most companies take advantage of this possibility. It is certainly an effective and reliable solution. However, it comes at a high cost. Small companies cannot afford this solution. Therefore, it is still popular to host PHP applications, although their speed is not the highest.

I tried the non-standard way. Since Microsoft promises to compile .Net Core also for the Linux operating system - I wanted to check it out. I found a provider that hosts up to two domains for less than \$30 a year. For such a low amount of money, one should not expect a lot of cladding power. For me, however, it was completely sufficient. The project is used to create an index of names found in the documents of the Polish Institute of National Remembrance, which I provided.

To create the backend I used Visual Studio 2022 Community version. I compiled the web application in .Net Core 6 version. I used MySQL as the database. The frontend is done in Angular. In this case, the appearance is not the most important, I used Bootstrap without any modifications.

You can try the application by clicking <https://teczka.men>. To read the basic data of the Linux server select Informcje and then Serwer from the menu.



## IPN - materials collected by the security service under communism

During the communist era, I was an opposition activist. The security service collected detailed materials on me in order to indict me. After Poland regained its sovereignty in 1989, access to the archives has been gradually opened.



I received my first materials from the IPN on March 13, 2004. A black-and-white photocopy of the contents of three folders stapled with colored string and sealed. My dear wife - Zosia laboriously scanned these several hundred pages. If it were not for her dedication to the stupefying task, you would not have, Dear Internet user, had the opportunity to get acquainted with a piece of communist Poland back in the early 21st century.

Like any makeshift, it survived the years and even reached adulthood. I made the current version in early 2023. I used Visual Studio Code as the editor. I used the Angular framework. I happened to be practicing its 15th version and made the new edition of the site as an exercise. Of course, the site was responsive. The HTML files were rife with `*ngFor` loops or `*ngIf` conditions. I made abundant use of directives, decorators and nested

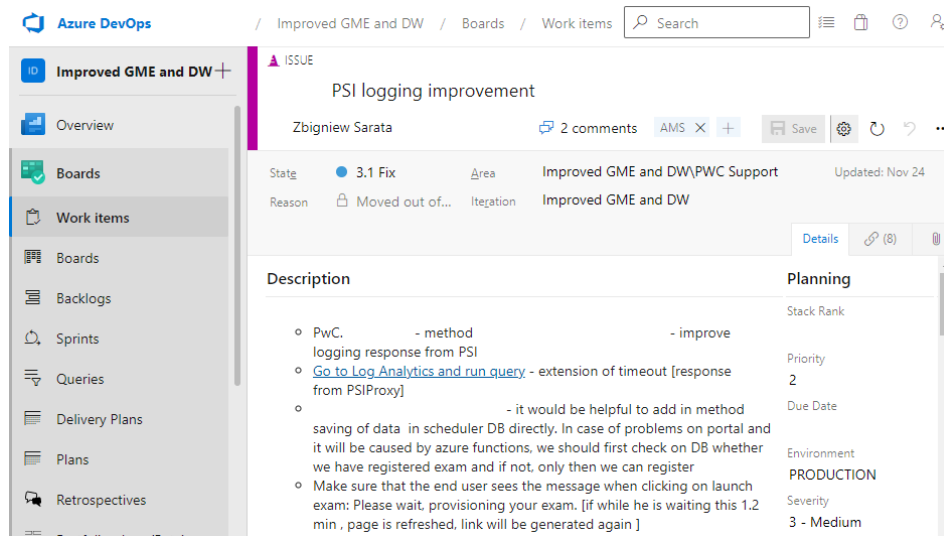
components. Of course I used routing and animations. See live at <https://teczka.sarata.pl>. and sources at GitHub.

---

## PwC – maintenance

In my PwC days, I worked in application maintenance teams. The supported solution included various components: mobile application, mobile application backend, portal, back office portal (with scheduling module), prediction/scheduling/reservation engines, Click Dimension, SharePoint.

The solutions were hosted on Azure Cloud platform and based on Microsoft Dynamics 365. Web applications benefited from Azure's rich ecosystem - there's a lot to play with - from Service Bus and Redis, to Azure Functions, SQL, Media Services, to App Inside and Log Analytics. From a DevOps point of view, I had an interesting cloud architecture here, with Dynamics in the CRM role.



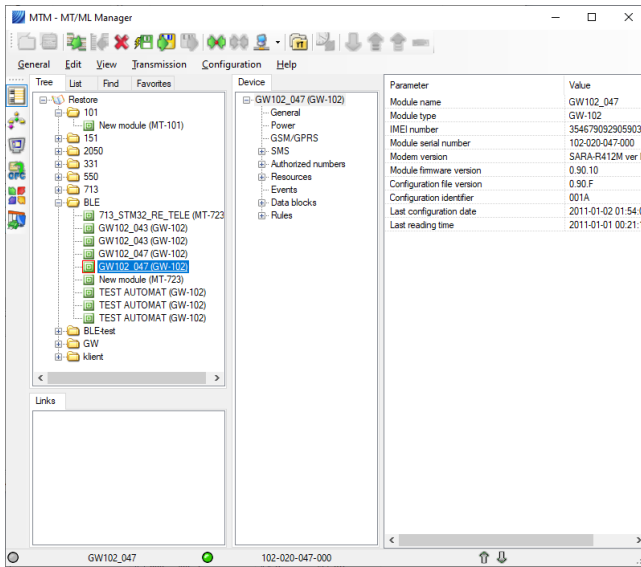
---

## MTM - telemetric modules manager

I started creating the application in the advent of .Net 1.1. The first functional component was the configurator of telemetry modules. I based it on XML files and XSD schemas. The language files were created in the XML standard too. XSLT transformation was also used in a few cases. So far, over 300 versions of configuration files for several dozen types of telemetry modules have been created.

Another functional components were also based on the XML standard. They were OPC configurator, PLC (mtProg, text programming, ladder programming), monitor, firmware update.

MTM is constantly developing/adapting to new functionalities of telemetry modules. One option is to support battery packs connecting to the server over GPRS usually once a day. In 2021, I added the Bluetooth LE interface.



MTM allows:

- Setting & getting module configuration
- Text PLC programming
- Ladder PLC programming
- View contents of internal registers
- Setting of OPC configuration
- Updating of modules firmware version
- Updating language messages

Communication modes:

- Cable: serial & Ethernet
- Remote GPRS over UDP frames
- Spooler for battery MT modules
- Bluetooth LE

Description: <https://www.inventia.pl/en/user-software-tools/>

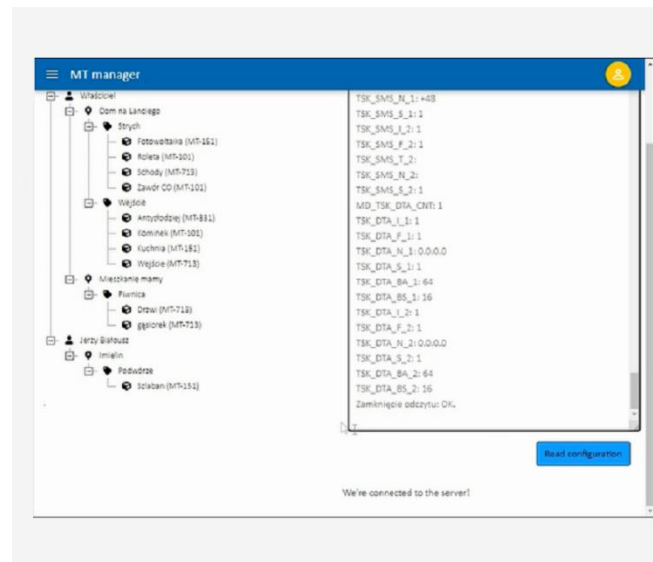
## mtmanager.net - web concept of MTM

In 2020 I implemented the web concept of MT manager. I based the backend on Node.js and MongoDB, the frontend on Quasar.js. This framework with Cordova and Electron support allows you to compile common JS code into native applications for Android, iOS, Windows, Linux and OSX operating systems.

The study has the normal ability to log in, register, remind, etc. Telemetry functionality is limited. My proposal was to share the module configuration with *friends* of module owner. The area drawings are 1024x1024. 128x128 module symbols.

The things I wanted to show are:

- Tree in the browser like in MTM
- The tree context menu
- Live preview of UDP transmission
- Serial port in the browser
- Json configuration files directly converted from xml with one click
- Compilation of installers for various operating systems



Demo: <https://youtu.be/bhdNr2Kc8dc>

## Agreus - agricultural production supervision and control system

Sometimes a mobile application contains limited functionality of a web application. But not with Agreus. The mobile application not only supports the user in the field, but thanks to the fact that the user carries the smartphone with him - it instantly shows important information and alarms

(Firebase Cloud Messaging). In the field, the user can indicate the azimuth and distance to the sensor, it allows, based on the location of the smartphone, to mark the area of cultivation of a specific plant on the farm plan. Near Field Communication allows you to read the serial number of the dirty sensor.



Agreus allows:

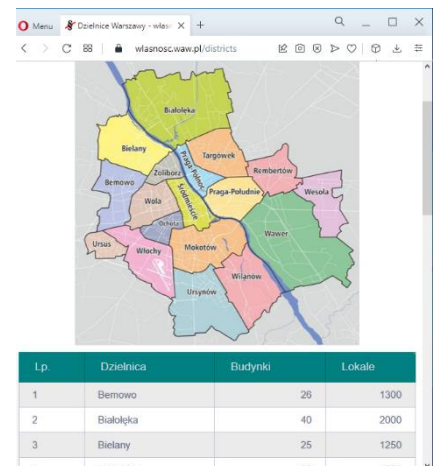
- Precise irrigation
- Precise fertilization
- Monitoring of climatic parameters
- Measurement of soil parameters and horticultural substrates
- Generating alarms and notifications
- Access to historical data

Description: <https://www.agreus.pl/>

## Ownership - social property project

The web application was created in 2016 out of the necessity to make an inventory of buildings stolen by municipal authorities. Residents paid for their construction, but according to the law, the owner of all apartments is the city of Warsaw. In the application, after entering the address, anyone can check whether it has been tainted. In addition, there is a list of such buildings for each district. Everything is created in PHP working with MySQL. Includes blog and CMS module.

Website: <https://wlasnosc.waw.pl>



## Xlsx file to MySQL database converter

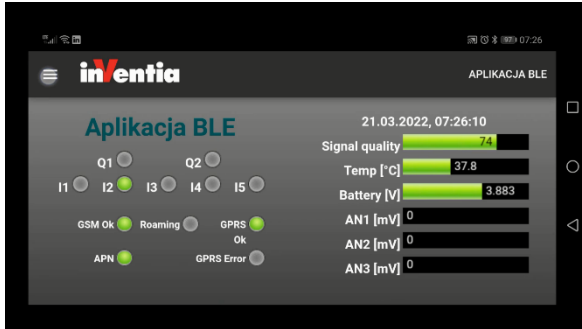
```
// -----
private void Wybory2014SQL(String sheet_name)
{
    Microsoft.Office.Interop.Excel.Range current_range = (Microsoft.Office.Interop.Excel.Range)ranges[sheet_name];
    if (current_range == null)
        return;
    HashSQL(wybory2014, wybory2014_id, null, "wybor2014", "terc_nr", null);
    for (int row = 1; row <= current_range.Rows.Count; row++)
    {
        try
        {
            System.Windows.Forms.Application.DoEvents();
            String terc_ = GetStringCell(current_range, row, "A");
            long id_gmi = (long)gminy_terc[terc_];
            Int16 okr_n = (Int16)GetNumberCell(current_range, row, "C");
            Int16 numer_ = (Int16)GetNumberCell(current_range, row, "D");
            String terc_nr_ = String.Format("{0}{1:d4}", terc_, numer_);
            String okr_nr_ = String.Format("{0}{1:d2}", terc_Substring(4), okr_n);
            if (!lobwody.Contains(terc_nr_))
            {
                MessageBox.Show("HT does not contain", terc_nr_);
                continue;
            }
        }
    }
}
```

The web application was created "pro bono" in 2017, when I was a member of the ".Nowoczesna" political party. Its task was to estimate the possible number of seats in parliament in future parliamentary elections. The calculations were made on the basis of the electoral law and the

actual voting results in the previous elections. A significant problem was converting data from Excel spreadsheets to MySQL database. The application was hosted in the Google cloud. The access to the app was limited to the ones who had a login from the party leadership.

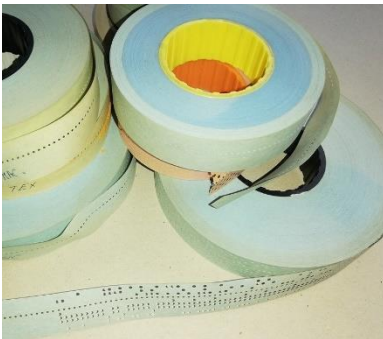


## Ble713 – mobile application



In February 2022, I created a mobile application dedicated to the MT-713 telemetry module. On the one hand, this application is a narrow subset of the MTM functionality, but it enables the presentation of the status of the module inputs, the status of counters, and the values of flow meters. It also enables configuration of APN parameters, 4G network and setting the module time.

## from punched tape to mobile applications



I was fortunate enough to witness the development of microcomputers from the very beginning. I started programming at the end of the 1970s. I lived behind the Iron Curtain, where new technologies did not reach. While some ICs could be smuggled out of the free world, it was impossible to buy the software utilities. The first assembler I had to write in machine language byte by byte. I saved programs on punched type. Only after that I was able to use the 8 "floppy disk.

It was not until the first half of the 1980s that the first IBM PCs arrived in Poland. The first unlicensed compilers of Pascal and C languages appeared with them. This made it possible to write application programs. The MS DOS operating system was king. However, even for the DOS system I wrote graphics programs limited by the resolution of the available monitors, first 320x200, then 640x400.

The widely available Windows operating system made it possible to use a graphical user interface. I created desktop applications. I have used a compiler from Borland and its Interbase for a couple of years. Then I switched to .Net from Microsoft and SQLServer.



Increasing the speed and reliability of the Internet along with the improvement of the MySQL database allowed me to write web applications in PHP language. Soon I was interested in Node.js and Angular. However, programming mobile applications was very difficult. Each manufacturer pushed its own solutions, unlike any other. The popularization of the Cordova overlay has finally opened the way to unification. The

latest framework that I use - Quasar.js allows you to create a web portal, application for Android, iOS, Windows, Linux and OSX operating systems from a single source .js.



Form over content - that's not my cup of tea. The functionality of the programs has always been a priority for me. I tried to propose some solutions to the end user, and I created the final version of the application in small steps in accordance with the currently modified assumptions. Only after some time did I find out that this method is called Agile.