DOI: https://doi.org/10.53052/9788367652018.10

Michał KUPCZAK[1], Marcin BERNAŚ[1], Vasyl MARTSENYUK[1],
Georgi DIMITROV[2], Dejan RANCIC[3], Oleksiy BYCHKOV[4]

INTELIGENTNY HUB PRACOWNIKÓW BIG DATA

Streszczenie: W artykule przedstawiono wdrożenie portalu Smart Big Data Job Hub opracowanego na podstawie wymagań projektu IBigWorld. Głównym celem portalu jest promowanie możliwości biznesowych pomiędzy uczelniami a firmami. Jako naturalną ewolucję portalu zaproponowano dwa rozwiązania: szablonowe i autorskie. Na koniec przedstawiono statystyki użytkowania i propozycję dalszego rozwój.

Słowa kluczowe: CMS, aplikacje webowe, frontend, backend, monolit modularny

SMART BIG DATA JOB HUB

Summary: The paper presents the implementation of Smart Big Data Job Hub portal developed based on requirements of IBigWorld project. Main goals of the portal are to promote business opportunities, between universities and business. Two solutions were proposed as natural evolution of portal: templated based and author solution. Finally, the usage statistics and further development plan was presented.

Keywords: CMS, Web applications, frontend, backend, modular monolith

1. Introduction

Project "Innovations for Big Data in a Real World" (iBIGworld) brought together higher education institution (HEI), business represented by companies and civil institutions to address the need of competence and job profile of employee needed on the market. The project [1-2] brings together four partners from four European countries: Poland, Bulgaria, Ukraine, Serbia. This collaboration provided innovative solutions to develop future Big Data experts. The prepared learning framework was based on IEEE guidelines for Big Data in Machine Learning". In this context the project aimed to not only create a bridge to fill the digital skills gap in southern Europe

[1] Department of Computer Science and Automatics, University of Bielsko-Biala, Poland: mk050666@student.ath.edu.pl, (mbernas,vmartsenyuk)@ath.bielsko.pl
[2] University of Library Studies and Information Technologies, Sofia, Bulgaria: g.dimitrov@unibit.bg
[3] University of Niš, Nis, Serbia: dejan.rancic@elfak.ni.ac.rs
[4] Taras Shevchenko National University of Kyiv, Kiev, Ukraine: oleksiibychkov@knu.ua

(according to European Commission's 'e-Skills for jobs' campaign) but also to build an ecosystem of key partners for creating an access port in underrepresented talent pools. This cooperation through the project phases allows to identify the underrepresented skills and find rationale behind the phenomenon of talented people, who lack the traditional credentials to land a good job in Big Data society. Therefore, a as a part of project the Smart Big Data Job Hub was established to promote business opportunities between universities and companies. The platform was intended to propose internship programs and PhD student supervision between HEIs and business. The platform is also meant as a medium to present ICT, Big Data and AI trends, new educational tools and learning resources. The proposed paper presents the technical issues concerning building the portal and possibilities of its further development trough sustainability phase.

2. Requirements

According to a project proposal, the institutions performing a project was aims to create Smart Big Data Job Hub that can fulfil project requirements specified in the proposal. Main goals of the platform are to promote business opportunities, through a Smart Big Data Job Hub, between universities and business. This includes new Big Data based internship programs, PhD student supervision between HEIs and business. Moreover ICT, Big Data and AI trends, new educational tools and learning resources are processed and disseminated through the Smart BigDataJob Hub towards the modernization of curricula. The Big Data Job Hub planned to be a place to disseminate the results of the project by presenting market needs and presents educational material. Project aims to provide all students and teachers with the opportunity to use digital tools, provided by the market and industry partners through the Smart BigData Job Hub, and to develop and upgrade their digital skills. Within a project it was assumed that projected hub will be:

- evolving employability requirements through the Smart BigData Job Hub, students will receive professional advice, support and experience, they will be able to build a well-oriented, in-line to the market needs specialization accessing increased employability opportunities,
- have the opportunity to respond to the rapid technology changes, through the outputs of this project,
- obtain state-of-the-art knowledge and skill background while accessing to the Smart BigData Job Hub will make job inquiring easier and appropriate on their expertise,
- co-supervision of master-thesis, open seminars, lectures, skill-aided tutorials and bootcamps,
- assuring for a gender- balance and non-discrimination, as outlined in the principles of ERASMUS+ CHAR.

According to agenda the users considering academics, students, researchers, professionals and industry managers of the Smart BigData Job Hub should reach 1000.

Based on the specified requirements the projected hub should consider with priority:

- automatic user registration (admin - possibility to delete) (priority high),
- 3 target groups of users (priority high),

- possibility of publication (text, images and documents) (priority high),
- links to the educational platform with the (educational) course (priority high),
- Search mechanism for the publication (priority high),
- Automatic information analysis (based on a dictionary of repeated terms) (priority low)
- Comment possibility (priority high)
- The cooperation mechanism (priority low).
- Information about visitors / active users (priority high).

Based on the requirements and development schedule two versions of platform was created. First one based on CMS and second as a dedicated solution.

3. Implementation of two solutions

First implementation is based on WordPress application [3]. The WordPress, as a CMS, offers a big flexibility in customizing the look and functionality due to vast number of elements and plugins. The CMS also allows to increase visibility in search engines by optimising separate webpages.

3.1 Instance based on WordPress

Installation of CMS requires defining core folders and uploading repository on web server. The folder that hosts all files are placed in public HTML folder. It is treated as root directory for WordPress website. The core files and folder's structure define the appearance, functionality, and security aspects of architecture. The service works based on PHP scripts and communicate with SQL based database. After installation the pre-installed default theme has to be change or modified. WordPress stores assets and theme files in dedicated theme folder directory. In this place the modification can be done. The back-end files are also vital for security reasons. The policy is defined for Apache server in .htaccess file. It's a critical WordPress core file that enables and disables the features of the website and certain plugins. The file is created automatically on root directory and it is usually hidden to minimize security risks. Second vital element (wp-config.php file) is responsible for establishing a connection between WordPress and the MySQL database. It should also be secured to minimalize possibility of unauthorised access to a database. Finally, all page content is stored in wp-content directory. This directory contains all plugins, theme files as well as all media and uploaded file (video, documents and image files). However, there is no structures that is related to the need of project. The file structure is flat. The solution is universal in its nature. To adapt solution to requirements the post plugin as well as statistic counter [5] was added by dedicated script in page template. Additionally, several security plugins were installed like wordfence. The configuration window is presented in Fig. 1.

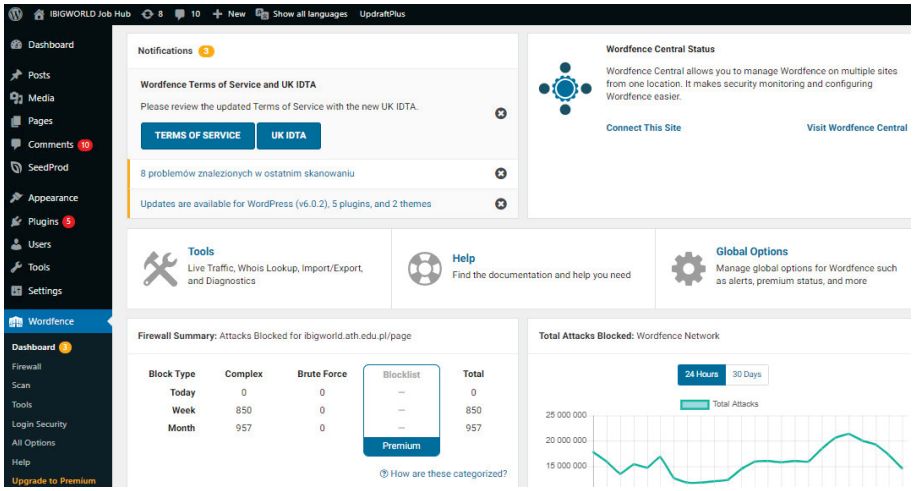


Figure 1. The representation of administration panel in current version of implementation

3.2. Instance based on authors implementation

The second instance is a part of a project, that was implemented as a part of a student engineering thesis and co-author of this work. Platform is separated to the backend API solution and the frontend application.

Backend

Backend is a .NET CORE API which allows the frontend application to receive necessary information from the platform database. Architectural solution is built using domain-driven-design (Fig. 2) and the whole solution may be considered to be a modular monolith (Fig. 3).

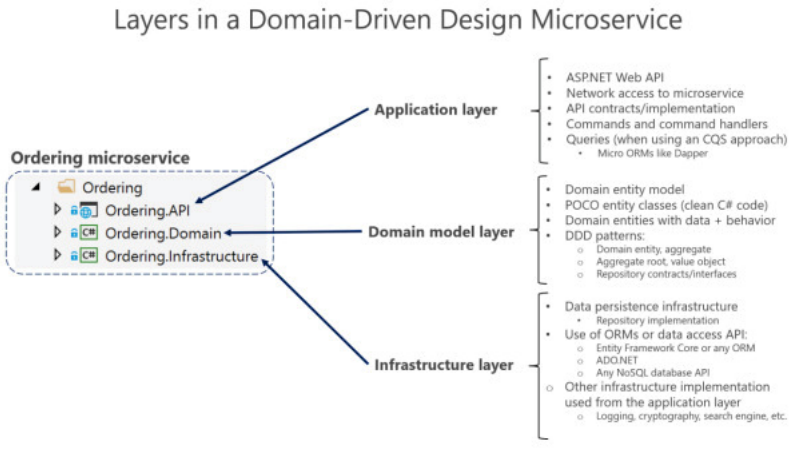


Figure 2. domain-driven-design example (DDD)[6]

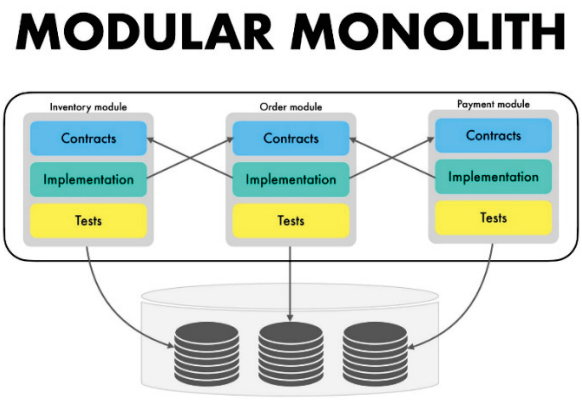


Figure 3. Modular monolith [7]

The authors modular monolith structure was proposed with one database. Due to the fact that the project seems to be moderate in terms of the size the author decided to use relational database MSSQL that allows developers to swiftly operate on the moderate number of relations and be sure that entities will not be created using a wrong relation (Fig. 4).

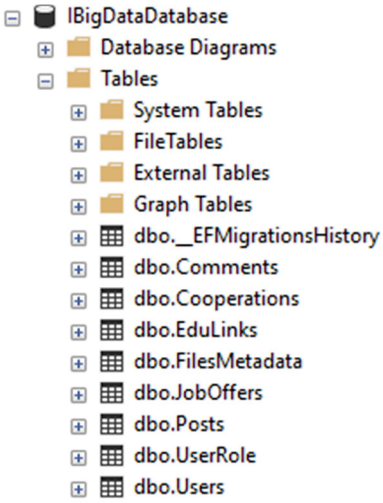


Figure 4. Database entities

As C# is used in .NET platform the natural approach was to use an object-relation mapping technique (ORM). In the end the author decided that the platform should consist of two ORMs - Entity Framework Core for building migrations, applying those to the database and managing the database, and the second one - Dapper for connecting and data extraction solutions. (Fig. 5) Using those two is very common for the command query responsible segregation (CQRS) architecture pattern that was applied for our solution, however in comparison to the traditional approach where the

commands use EF Core, our system rely only on dapper in terms on executing database commands and queries, nonetheless it is worth considering using ef core on the commands in the future as a result of latest performance updates in .NET 7 (Fig. 6).

Fig. 5. Example usage of dapper within query module

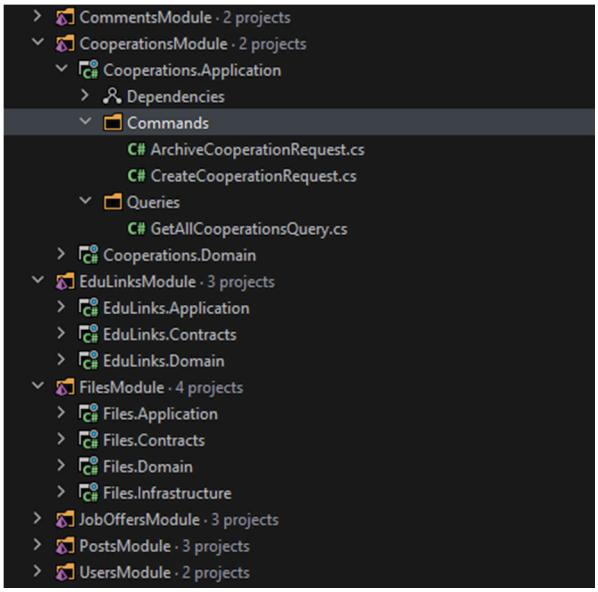


Figure 6. Authors implementation of modular monolith, DDD and CQRS

Every endpoint that requires security access are secured using resource-based approach. (Fig. 7).

Fig. 7. Example of an endpoint secured using resource-based authorization

The solution also consists integration tests that are build using xUnit library. The folder is kept separately from the modules as it is intended in integration testing to test the whole endpoint flow rather than methods and units so it makes little sense to separate tests among folders. For this moment testing the endpoint health status by checking the endpoints and their results is more than enough for the solutions needs (Fig. 8).

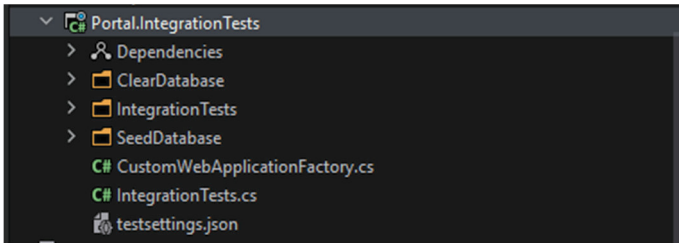


Figure 8. Integration tests solution

Frontend

The application is a single page application (SPA). Before the first line of code was written the author decided to use the react library that massively improves and simplifies the user interface development process. Usage of react allows the developer to build page using reusable components, so the whole development process is much shorter. The code is written mostly in typescript, so it is much easier for the developer to keep the code clean. Furthermore, due to the high regard of the SOLID and DRY rules, the solutions enforce specific styles and lint principles that takes care of potential styles and logic inconsistencies. The author also decided to separate the logic from the visual components (Fig. 9).

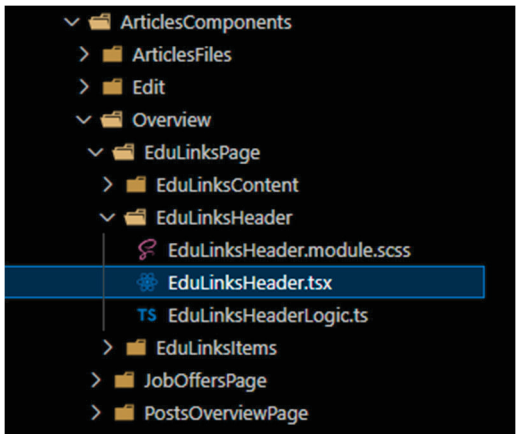


Figure 9. Application files logic separation structure

For styling the author decided to use CSS with pre-processor (SCSS) so the styling offers much more to the developer than the plain CSS could. For communicating with the backend API, the author uses axios library. The frontend is partially cached and is build using lazy loading principles to reduce the overall bundle size (Fig. 10).

```
const RequestRolePage = React.lazy(
  () => import("pages/RequestRolePage/RequestRolePage")
);
```

Figure 10. One of lazy loading implementations example

Due to the fact that there is some complex business logic on the frontend unit tests were involved here. In the nearest future E2E tests will also be implemented.
The design for the platform was created entirely by the author in a tool called figma (Fig. 11).

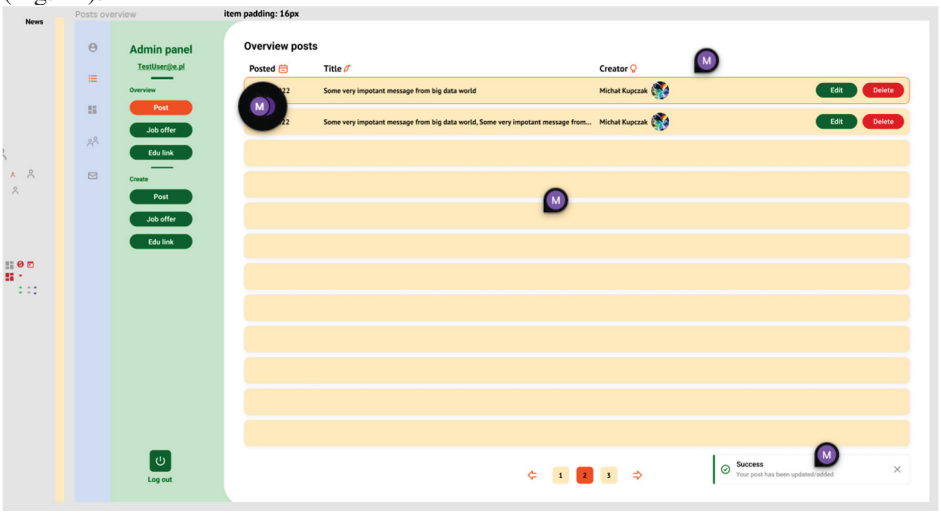


Figure 11. Prepared design example.

Deployment and clouds

The author uses the GitHub actions CI/CD. For this reason, before deploying the code to the cloud the developers can enforce tests and build workflows to run in order to reassure that everything will be working fine on the production. There are a few workflows that are enabled on committing code onto the pull request: One for cloud database and the second for tests (Fig. 12).

Figure 12. Github actions CI/CD

Database is hosted on the azure cloud and it uses the lowest CPU units possible on the azure in order to keep the costs of the databases low. Every time the workflow runs, the migrations script is created and then applied to the cloud database. The second one checks that every test in the frontend application works as intended. After the person has decided that every flow works well and the code looks good, he triggers the merge mechanisms that run the API and frontend app build and deploy workflows. During those processes the API is built on docker and then deployed to the google cloud and the frontend application is build and then pushed as static files to google cloud storage. In the nearest future the author will create flows for integration tests flow and the cypress tests flow to run on code pushes.

3.3. Comparing of solutions

The solutions were analysed taking under consideration multiple factors that are essential through a lifetime of the application [4].

Price

The most considered factor for the WordPress is price. The dedicated solution may use many different solutions that may cost much, and optimizing costs for the dedicated solution is one of the most important factors that can improve the dedicated CMS usability. However, it is worth to note that custom solutions require custom solutions providers and the price should be considered as a disadvantage for dedicated solutions.

Maintenance

Due to the fact that the WordPress is trying to focus on small or medium size project it is considered relatively easy to maintain the solution on WordPress. However, the dedicated solutions are not considered hard to maintain unless at least mid-experienced developers work on it and supervise the general architecture of an application. For this reason, even a large-scale project that are not a WordPress focus may be handled with ease when the appropriate specialists work on it.

Purpose

WordPress has limited use cases for building sites. It is limited to blogs, small pages and etc. However, one can create almost every type of application for the custom-made solutions. Even during the development process the goal may change and the application may be easily adapted to the required solution.

Development

Development should be definitely considered as an advantage of dedicated solutions. WordPress is limited to a certain number of plugins that mainly focus on small, medium size pages. For this reason, developing a large-scale project with custom integration and solutions on WordPress does not make any sense. In a dedicated solutions the only limit is our imaginations and the developers' abilities to create something special.

Plugins/libraries

There are limitless plugins and libraries in custom solutions. It depends on the platform one choose to build our project on. Not to mention that there are multiple specialized languages and platform for different application use-cases. For this point the most difficult part is for the developer to know what kind of solution will suit his needs the best. The WordPress has dedicated plugins and libraries that allows the users

to use those with ease. However, there are numerous limitations when designer want to attach some kind of special functionality.

Security

Due to the fact that different sites may handle different types of attacks it is obvious that WordPress has some dedicated solutions that may be considered comprehensive. For the dedicated solution designer are not limited to the standard security system and it may apply custom ones that fit exactly our solution, and remove redundant ones.

Time and human resources

Due to the fact that WordPress is a dedicated solution for small projects, the current platform would be definitely faster implemented there. Dedicated solution often requires time to think about possible solutions, and then the development team has to decide what kind of solution will fit their needs the best. However, the experienced development team will be able to achieve much betters results for the custom-made solutions.

Errors handling

WordPress is a dedicated system that handles errors well. When designer have a dedicated solution errors may be much more unexpected and it have to consider the fact that the whole solution that the development team is building may be worthless due to lack of appropriate research. This also may be a problem when unexperienced team will try to implement a custom-made solution.

Updating system and latest changes

Custom solutions can adapt new solutions faster than WordPress that relies on stability and reliability. There will be much bigger gap between implementing dedicated solutions for custom made application and WordPress solution. It is totally up to dev team when the update will be carried on in contrary to the WordPress where the team will have to implement those changes into their system.

Summing up, the proprietary CMS content management system is designed for larger projects, while the pages on WordPress have basic or intermediate features. Therefore the WordPress was used at the first stage of hub development and naturally it can migrate to proprietary solution when its scale will rise.

Moreover, WordPress should be chosen when developers have little to no experience. In other cases when designer want to create something special, or a system that may be more comprehensive, complex it should definitely stick to the custom-made solutions.

4. Presentation of implementations

The first implementation is in use and it provides a platform for communication and exchange information. Using post plugin each registered user, with at least contributor rights, can create an information backed up by data, documents or media file. Portal is operational and multiple various posts was placed from various sources: HEI, business or students. The portal currently presents various types of information categorised by tags. The example of specific tags posts headers is presented in (Fig. 13).

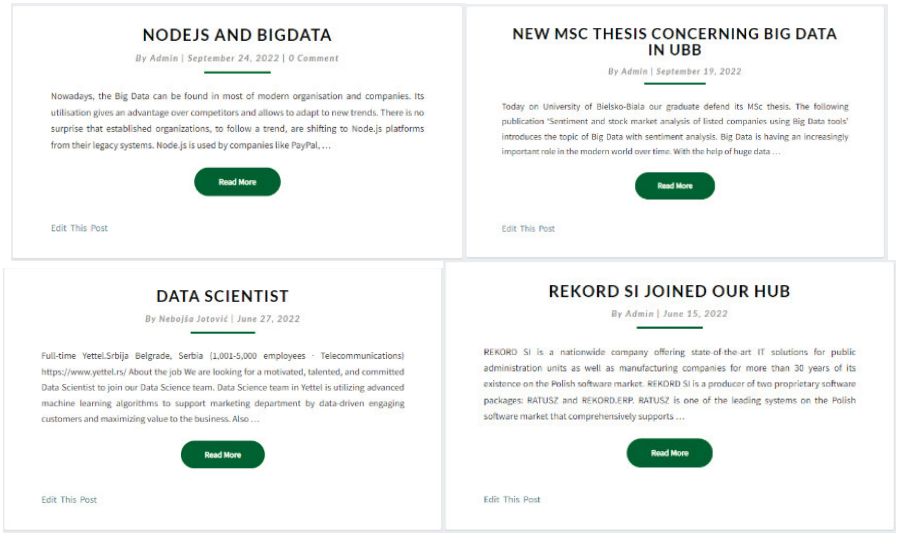


Figure 13. The example posts headers information available on portal[8-10].

The full post has a form of a web page, where both media and documents can be attached. The example is presented in Fig. 14.

Figure 14. The post containing project meeting.

Despite the popularity of a portal, its functionality is limited only to those available by WordPress plugins. Therefore, the authors implementation was proposed to available migration of portal to new standards, if the current tool proved insufficient in time.

4.2. Functionalities of authors solution

The most important functionality for the platform was the login solution. For the security reasons and the complexity of the problem it is considered a good practise to use third party solutions to keep the project safe. The author decided to use the Auth0

solution that provides a complete flow for users to log or register into the platform. Due to the fact that Auth0 is also a free solution till 7000 users' logins per week and it provides multiple login solutions, it was a straightforward choice (Fig 15).

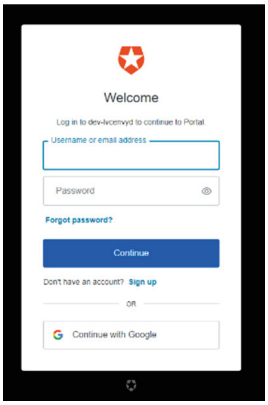


Figure 15. Portal Auth0 login solution.

During the process, the user may log or register in. There is also an option to use third party authorization like google login. After the user provides required data, the request goes to the auth0 solution that then returns a token to the application. The token may be later used to validate the API requests, so there will not be unauthorized calls to the backend. On the backend every time the requests come there is a middleware that runs just before the endpoint starts that checks if the user was registered in our database and if the user data form a token is valid. After the successful validation, the process of calling backend continues (Fig. 16).

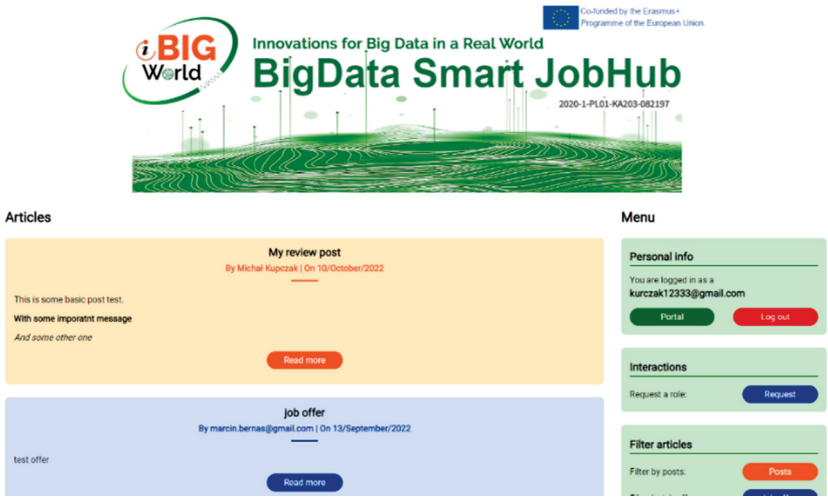


Figure 16. Portal view after successful login.

The main purpose of the platform is to provide users an overview of the latest information given by the IBigWorld community. There are 3 types of articles that can be created in the administration: Posts, Job offers, Edu links. There are different roles

in the system, each role has different set of permissions. The administration can manage the platform by entering the portal, not registered - under no circumstances can enter administrating portal as well as the standard users (Business/Student). The employee and HEI rank allows to enter the portal management and the users can view add edit and delete only his posts. The posts may be commented by users. Only the administration can overview and edit every comment. The user is able to filtrate the posts on the main page (Fig. 17).

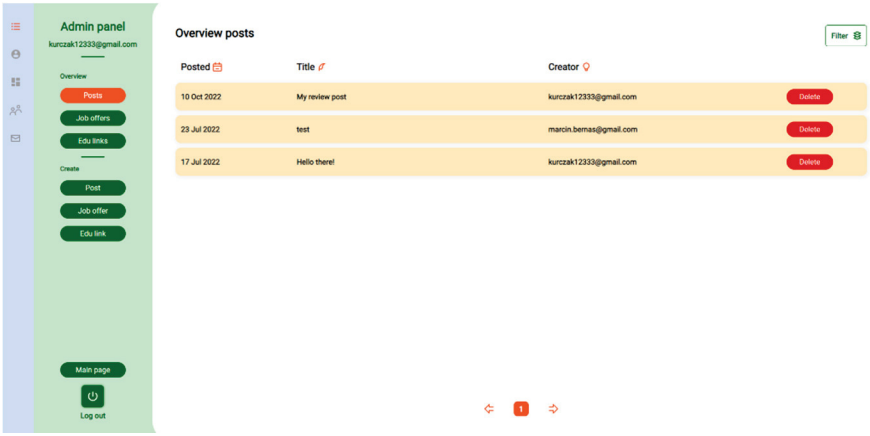


Figure 17. Administration portal overview.

The posts can consist document and images. The solution uses the google cloud storage where the files are placed after the user has uploaded one to the system. During the process the application store the metadata of the image in order to get the images quicker (Fig. 18).

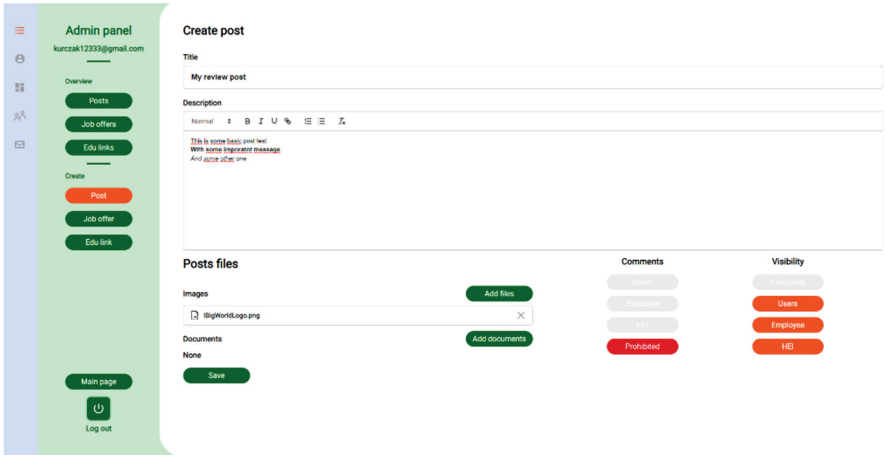
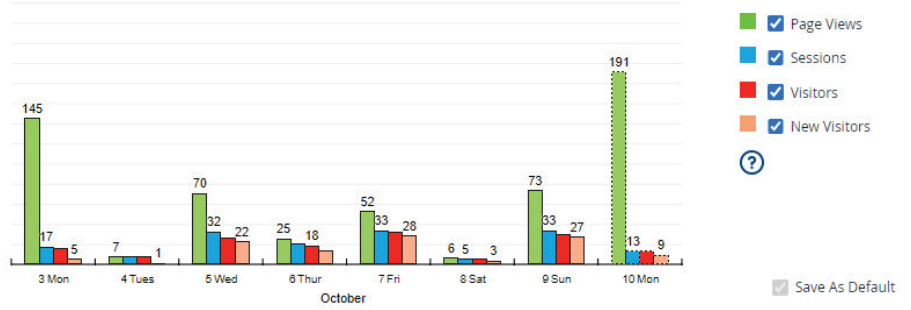


Figure 18. Creating post page.

To request a role in the system, users can send a request to the portal administration with a note that informs the administration of his goal.

4. Statistics

The proposed hub in first version become operational from second phase of a project, however the full statistical analysis was included, when traffic started to grow over 100 visit a day. Therefore, the statistic is presented from second quarter of 2022 year, when the multiplier events promoting solution took place. The statistics are made by external company stat counter to increase its robustness. The result of traffic is presented in Fig. 19.



Average Daily Page Views	Average Daily Sessions	Average Daily Visitors	Average Daily New Visitors
71.1	20.0	18.3	13.5

	Page Views	Sessions	Visitors	New Visitors
Mon, 10 Oct 2022	191	13	13	9
Sun, 9 Oct 2022	73	33	29	27
Sat, 8 Oct 2022	6	5	5	3
Fri, 7 Oct 2022	52	33	32	28
Thur, 6 Oct 2022	25	20	18	13
Wed, 5 Oct 2022	70	32	26	22
Tues, 4 Oct 2022	7	7	7	1
Mon, 3 Oct 2022	145	17	16	5
Total	**569**	**160**	**146**	**108**

Figure 19. Results of users' usage of platform in within a week [5].

As it was presented in Fig. 19 the traffic on the web page is equal to 90 visits on average per day. It is worth to note that stronger traffic can be noticed every Monday. This could indicate that some users start a week with search of news on the platform. This could be used to plan the information updates on a site. It is also worth to note that weekly a number of new users are growing significantly (108 new users last week).

The general statistic shows that, at the end of the project, over 1000 new users visited a site and it is meaningful to maintain it and add new information at least base on weekly basis. Statistics shows, that the portal is popular among the countries, which were participants of the project (Fig. 20), however new countries start to appear in rankings (e.g., Australia).

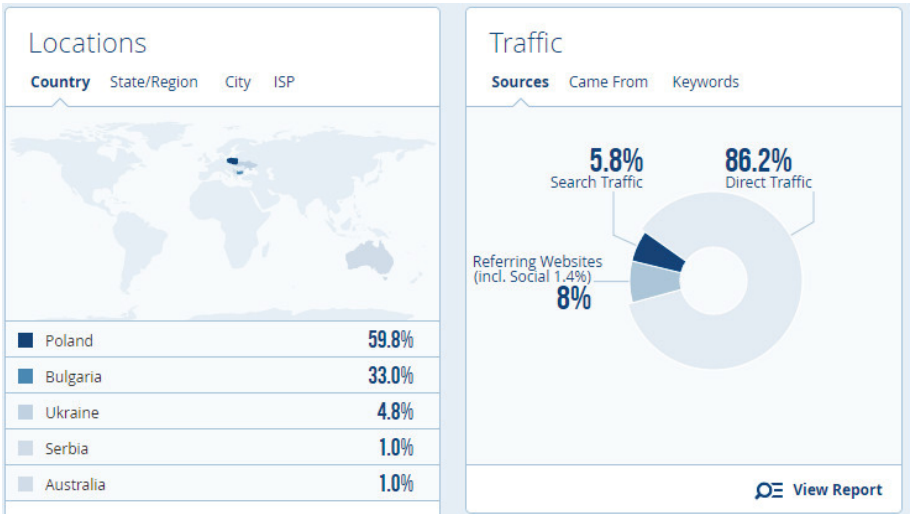


Figure 20. The recent activities on the page [5].

The overall visit values are presented in Fig. 21. As presented in Fig. 21 up-to-date 1319 new users visited a site and the portal was opened 124 thousand times. It is expected that after promoting the result of project the number can still increases to stabilise at assumed by the project level. In case of rapid increase in portal visits in the future the migration to new version become justified necessity.

	Page Views	Sessions	Visitors	New Visitors
+ Q4 2022	934	191	150	122
+ Q3 2022	123,201	1,410	1,154	1,148
+ Q2 2022	331	116	49	49
Total	**124,466**	**1,717**	**1,353**	**1,319**

Figure 21. Overall statistic for user views.

5. Sustainability and further development

The solution met acceptance of the users and the number of portal users is growing. This is a motivation to keep constant updates to a portal. Statistic results suggest that updates should be done especially before each Mondays. The current version of platform proved stable, however if the number of users will grow the platform will be migrated to developed authors solution. There is no shadow of a doubt that the solution should be reliable. For this reason, before migration the proposed authors 'solution will be enhanced by some testing mechanisms that will improve the current system test coverage i.e.: xUnit integration tests on BE and Cypress tests on FE. The author also considers migrating the react solution to the next.js one because it has

some more caching opportunities than react and the main page could be run much faster. There are also other functionalities that are planned but limit of time restricts the current development time from proceeding further. The last problem is that the requirements for the platform have little supervisor, so it is hard to build a solution having only a major features requests – the development process is often a complex one and different obstacles occur unexpectedly. If there were interest in the platform there could be a better design implemented, and other more interesting features as well.

REFERNCES

1. The project webpage *https://erasmus-plus.ec.europa.eu/projects/search/details/2020-1-PL01-KA203-082197* (access 10.10.2022).
2. iBigWorld project: *http://ibigworld.ni.ac.rs/* (access 10.10.2022).
3. MESSENLEHNER B., COLEMAN J.: Building Web Apps with WordPress. WordPress as an Application Framework. O-Reily. ISBN: 978-14-919-9003-2, 2019.
4. Online web-page: *https://crafton.eu/blog/wordpress-or-dedicated-cms*. (access 10.10.2022).
5. The traffic statistics portal. *Https://Statcounter.com* (access 10.10.2022r)
6. Designing DDD oriented microservces *https://learn.microsoft.com/en-us/dotnet/architecture/microservices/microservice-ddd-cqrs-patterns/ddd-oriented-microservice* (access 10.10.2022).
7. Modularising the monolith *https://speakerdeck.com/avosalmon/modularising-the-monolith-laracon-online-winter-2022* (access 10.10.2022).
8. The example to student information *https://ibigworld.ath.edu.pl/index.php/en/2022/09/19/new-msc-thesis-concerning-big-data-in-ubb/* (access 10.10.2022).
9. The example of HEI post: *https://ibigworld.ath.edu.pl/index.php/en/2022/09/30/erasmus-project-ibigworld-transnational-meeting-m4-has-been-conducted-at-the-university-of-library-studies-and-information-technologies-in-bulgaria/* (access 10.10.2022)
10. The business post example: *https://ibigworld.ath.edu.pl/index.php/en/2022/06/15/rekord-si-joined-our-hub/* (access 10.10.2022).