

Bachelor Thesis

Computer Science and Engineering, 300 credits



Company Intranet

Thesis in Computer Science and Engineering, 15 credits

Halmstad 2021-09-06

Abboud Afram, Danial Sarab Fard Sabet



Acknowledgments

We would like to express our sincerest gratitude to our supervisor Hossein Jeddi Abdarloo for his great help with the report. For setting up meetings, giving feedback, answering all our questions, and always being available. We also want to thank our examiner Veronica Gaspes for all the feedback which improved our bachelor thesis.

Abboud Afram & Danial Sarab Fard Sabet
Halmstad 2021

Abstract

Many companies use intranets where only authorized personnel can share, create, and access news, events, and knowledge in a company. An intranet is a web application with dynamic content. It consists of various functions and features that facilitate the information management and knowledge flow within the company. The main goal of this project is to create and design an intranet demo that is adaptable and extendable by companies and organizations. The intranet contains various primary functions that users can interact with. The designed intranet is based on the EPiServer Content Management System framework and is programmed in back-end and front-end with the Visual Studio IDE tool. The back-end part is mainly programmed in C# language to create functions and logic such as the intranet structure and then linked to the front-end part where the HTML, CSS, and JavaScript library React is used to encode the front-end and get a user interface. The finalized intranet contains various primary functions where authorized employees have individual profiles with their status and contact information. Users can share corporate events and access corporate news.

Sammanfattning

Många företag använder intranät där endast auktoriserad personal har tillgång till att dela, skapa och komma åt nyheter, händelser och kunskap inom företaget. Intranät är en webbapplikation med dynamiskt innehåll. Den består av olika funktioner som underlättar informationshantering och kunskapsflöde inom företaget. Målet med detta projekt är att skapa och designa en intranätdemo som är anpassningsbar och extensibel för företag och organisationer. Intranätet innehåller olika primära funktioner som användarna kan interagera med. Intranätet är baserat på EPiServer Content Management System ramverket och är programmerat i back-end och front-end med Visual Studio IDE-verktyget. Back-end delen kommer huvudsakligen att programmeras i C# språk för att skapa funktioner och logik som strukturen av en färdig designat intranät. Sedan länkas färdig programmerade back-end delen till front-end delen där HTML, CSS och JavaScript-biblioteket React används för att koda front-end för att få ett användargränssnitt. Den utvecklade och slutförda intranätet innehåller olika primära funktioner där auktoriserade anställda har enskilda profiler med sin anställnings status och kontaktinformation. Användare kan dela företagsevenemang och få tillgång till företagsnyheter.

Table of Contents

| | |
|---|-----------|
| Acknowledgments | 1 |
| Abstract | 2 |
| Sammanfattning | 2 |
| 1. Introduction | 5 |
| 1.1 What is an intranet? | 5 |
| 1.2 Problem definition | 5 |
| 1.3 Purpose | 6 |
| 1.4 Goals | 6 |
| 1.5 Restrictions | 6 |
| 2. Background | 7 |
| 2.1 Web development | 7 |
| 2.1.1 Back-end and front-end | 7 |
| 2.2 CMS technology | 7 |
| 2.3 Analysis of the various CMS tools | 8 |
| 2.3.1 Learning phase and support | 8 |
| 2.3.2 Plugins | 9 |
| 2.3.3 Cost, Safety, and Security | 9 |
| 2.3.4 Popularity and trendiness | 9 |
| 2.3.5 Developer experience | 9 |
| 2.3.6 Summary | 10 |
| 2.4 EPiServer CMS | 10 |
| 2.4.1 EPiServer Search & Navigation | 11 |
| 2.5 Tools and languages | 11 |
| 2.5.1 ASP.Net | 11 |
| 2.5.2 C# | 12 |
| 2.5.3 JavaScript with React library | 12 |
| 2.5.4 HTML and CSS | 12 |
| 2.5.5 Visual Studio | 13 |
| 2.5.6 GitHub | 13 |
| 2.6 Types of intranets | 13 |
| 3 Method | 15 |
| 3.1 Task Specification | 15 |
| 3.2 Starting up | 15 |
| 3.3 Data access layer | 16 |
| 3.3.1 Pages | 16 |
| 3.3.2 News and events feed | 19 |
| 3.3.3 Tags and restrictions | 19 |
| 3.3.4 People List | 20 |
| 3.3.5 Services and dependency injection | 21 |
| 3.3.6 API Controller | 21 |
| 3.4 Presentation layer | 22 |
| 3.4.1 Design | 22 |

| | |
|---|-----------|
| 3.4.2 Requests | 23 |
| 3.4.3 Routes | 23 |
| 3.5 Result analysis | 24 |
| 4 Result | 25 |
| 4.1 Pages | 25 |
| 5 Discussion | 31 |
| 5.1 Company implementation of our intranet | 31 |
| 5.1.1 Social impact | 32 |
| 5.1.2 Safety and security | 32 |
| 6 Conclusion | 33 |
| 7 References | 34 |

1. Introduction

In the early 90's, intranet technology was introduced into the world, which emerged as the application of internet technologies to business purposes [1]. Intranet technology is a private computer network that uses internet protocols. It is restricted for employees within a company or organization to improve the work environment [2].

1.1 What is an intranet?

The amount of information and data is increasing within organizations and companies. Firms are continually looking for technology forms and strategies to manage their knowledge. An intranet is an example of such technology used by many organizations and companies because it can be regarded as information and a strategic tool in knowledge management [1]. Communication within companies can be a sensitive and complicated topic. This is partly because most large companies consist of offices distributed geographically and multiple departments containing various types of important information [3,4]. Information and data have great value and importance. This information and data content needs to be transferred, secured, and accessed by authorized users [5]. Therefore, it has become increasingly important to structure information and make it more accessible to those who use it.

An intranet is a web application tool that helps companies create and share information in a secure way. It allows organizations and companies to incorporate real-time communication capabilities [6]. An intranet will let organizations and companies gather all the information and data in one place, which will help to bridge any segmentations in the business. Intranet is divided into three central definitions. First, the intranet runs typical internet applications and is based on TCP/IP protocol. Second, it is a private network which means that access needs to be granted on a selective basis. Third, the intranet does not address any specific, well-defined need compared to the traditional intra- organizational information systems [7]. The development of an intranet can vary depending on the organization and on multiple factors, including the size and type of the organization. Intranet is mainly implemented for organizational needs to allow geographically distributed employees to share, create, and access corporate information [5].

By implementing the intranet as an information and communication technology into the work environment, colleagues will connect, generate, and integrate knowledge, and the knowledge flow will be facilitated [8]. The technology will allow the community experts to develop, which is vital within firms, by letting employees share, brainstorm, and receive feedback in various discussion groups [5]. Intranet is a central hub for the company online, different conversations will occur in the hub, and company events are planned together with shared company news and much more [9]. The technology is the hive of activity and will improve engagement in several ways by letting the employees understand the bigger picture and feel part of the company's culture.

1.2 Problem definition

This project is dedicated to summarizing knowledge and methods to develop an intranet. Research and studies are crucial to understand how to apply web development for creating an intranet, analyze which functions an intranet should contain to be usable for companies and organizations, and examine how intranets are structured as a web application.

1.3 Purpose

The purpose of this project is to experiment with software development for web applications to adapt and create our own intranet demo based on the EPiServer CMS framework with primary functions and develop a UI that can be adapted and extended by companies and organizations.

1.4 Goals

This project aims to create our own intranet containing multiple primary features and functions compatible with any company. The goal will be divided into three sections: The back-end programming, which contains an analysis of the web publishing tool EPiServer and programming with C#. The front-end programming will include the development of the design with JavaScript and React library. The last part will be combining the two major parts to get a functional intranet. Implementing and developing our own User Interface (UI) and adapting a user-friendly design will facilitate the navigation to acquire information within the intranet [10]. Depending on time, our intranet will consist of these most implemented functions:

- *Pinboard*
Allows employees to post their news, pictures, activities, etc.
- *News and events*
News within the company and the employees can be shared and quickly changed to keep everyone updated. An event planner that allows everyone to plan a work-related event.
- *File exchange*
According to the company's standards, files and documentation can easily be exchanged between departments and employees and saved on the company cloud.
- *User profile and login*
Allows employees to create an individual profile depending on their role in the company.
- *Search*
To quickly navigate within the intranet.

1.5 Restrictions

This thesis will be restricted by the following factors:

- The main focus of this project will be on the software development of dynamic web applications.
- Due to the time limit, the intranet will consist of primary and practical functions, which will affect the quantity and the quality.
- Because developing web applications within the software field is broad and diverse, the societal issues will not be discussed deeply. Instead, the time will be concentrated on gathering development knowledge and developing the product.

2. Background

In this section, the various tools, theories, and research will be presented together with the development of web applications. This will include how intranets are designed and developed using EPiServer CMS and ASP.NET frameworks, how EPiServer CMS simplifies developers' work by offering valuable functions and easy navigation in Edit-Mode. Moreover, this section will give a better understanding of various programming languages, how they are written and used with the help of Visual Studio application tools, and how code is shared using GitHub. Various tools for creating an intranet will be presented and compared, also why EPiServer CMS was chosen. There are different types of intranets, and these types will be presented for a better insight into the created intranet and into software development and web design.

2.1 Web development

Web development contains various aspects such as web design, web publishing, and database management. These aspects are the structure of creating, building, and maintaining websites. Different CMS:s are used to develop websites. CMS contains tools that allow web developers to create and edit dynamic websites using a web-based interface. In this project, EPiServer CMS is used [11]. Web development consists of two parts: back-end and front-end programming, which are essential for all dynamic websites. Both back-end and front-end work together to achieve a completely functional application. Some organizations divide the work into development teams to create separate front-end and back-end. During this project, both back-end and front-end are programmed, which is known as full-stack development [12].

2.1.1 Back-end and front-end

The back-end programming is defined as the data access layer and is a layer that establishes communication between the database and the back-end service. The fully developed back-end part is not visible to users. The back-end part contains functions and operations processed before integrating with the front-end and displaying the website's content. On the other hand, the front-end programming is defined as the presentation layer and is the website's UI. The front-end includes multiple elements such as the application layout, graphics, text content, etc. The input from the user is received through the front-end and processed in the back-end. The back-end handles the request, manages the data, and sends outputs to the user through the front-end [12, 13].

2.2 CMS technology

CMS are data repositories that provide advantages like improved information accuracy, increased flexibility, enhanced system management, and reduced maintenance and cost [14, 15]. It is built on top of web application frameworks and allows websites to add, delete, and update the content without changing the code [16]. The main focus of CMS is to manage websites, and they are crucial to managing and delivering business knowledge. CMS is a tool that helps with building websites without the need to start from scratch. This is possible by handling basic infrastructure like creating web pages, storing images, and other functions. Depending on which CMS is used, the web application framework is different. A CMS consists of four foundations, the database, the application programming interface (API), UI, and the views, as shown in Figure 1.

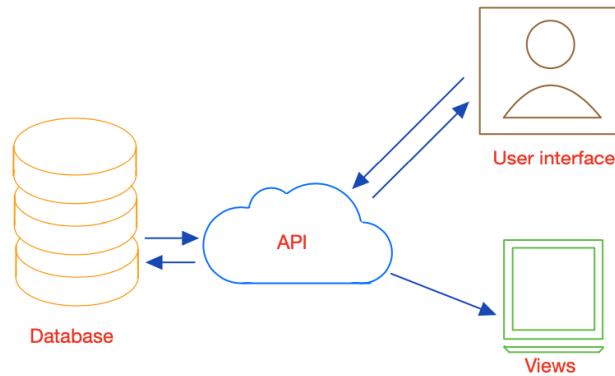


Figure 1. *The four foundations of CMS [17].*

The data is stored in the Database. Editors and administrators use the UI to access and manage the data in the database. The templates of the views determine the displayed content and the layout of published data. The API consists of an external part and an internal part. It is an interface between the database and the UI. CMS uses the internal part and cannot be touched by editors, while developers can use the external part for customizing their solutions [17]. EPiServer is one of the options to develop an intranet, and there are lots of tools and a variety of CMS: s to use. In section 2.3, WordPress, Drupal, Umbraco, and EPiServer will be compared in the following aspects:

- Learning phase and support
- Plugins
- Cost, safety, and security
- Popularity and trendiness
- Developer experience

This comparison will give the reader an understanding of why EPiServer CMS is used for this project.

2.3 Analysis of the various CMS tools

In this section, different CMS such as WordPress, Drupal, Umbraco, and EPiServer are compared in different aspects mentioned above. While each one has its advantages and disadvantages, we discuss the reason for choosing EPiServer as the CMS in this project.

2.3.1 Learning phase and support

WordPress uses MySQL database and is written in PHP. The CMS has documentation that is easy to follow as a non-experienced developer with great support and fast learning curve, which makes the CMS more user-friendly and popular. The users of WordPress are not required to have a lot of experience in web development to use the CMS [18, 19]. Similar to WordPress, PHP is the programming language used in Drupal with MySQL as a database. But the Drupal CMS has a more advanced learning curve with good community help and support. It requires their users to have knowledge and experience about web development. Drupal CMS is more suitable for developers than regular users. Overcoming Drupal's advanced learning curve will allow developers to create larger and more advanced projects, but the websites take time to load due to the large breadths of tools in the CMS [18, 19, 20]. Similar to Drupal, Umbraco also has an advanced learning curve and is dedicated to experienced developers. This is due to the lack of documentation and the use of the .NET framework. Umbraco is written in C# language and uses Microsoft SQL database. Umbraco offers its users a helpful community to overcome some of the difficulties [18, 21]. EPiServer CMS, same as Umbraco, is built on .NET framework, written in C#, and uses the Microsoft SQL database. Similar to the other CMS, EPiServer has an advanced learning curve and is suitable for experienced developers. But the CMS

offers their users commercial support, forums, and well-documented guides, which facilitates the learning phase and the usage of the CMS [18, 22].

2.3.2 Plugins

WordPress is initially designed as a blogging platform, and many plugins are one of the main advantages [23]. The CMS consists of many plugins developed and released by independent developers. This makes the website more insecure and outdated, which leads to the risk of being hacked [18, 24, 25]. Plugins are also necessary for Drupal that can be found free or purchasable [18, 26]. Umbraco only requires plugins for extra features for the website, and it offers its users a good number of free plugins [21]. EPiServer has a more extensive plugin system for extending and adapting the CMS UI. The CMS provides extra modules that can be integrated easily to add quick enhancement to the website [21, 23].

2.3.3 Cost, Safety, and Security

WordPress CMS is an open source with purchasable plugins. Each plugin has a risk when installed; due to a large amount of independently developed plugins, the website can become outdated and insecure. The CMS allows unlimited login attempts, making the website an easy target for hackers [18, 24]. Drupal is also an open source CMS but compared to WordPress, the CMS offers strong security features with less prone to hacking. Drupal offers features such as access and user permissions that allow the user to create roles with individual permissions. [18, 26]. Umbraco is one of the most secured CMS on the market and is open source [27]. The CMS has a built-in program that allows the Umbraco team to deal with security issues before they occur. It undergoes regular penetration testing and third-party inspection [26]. EPiServer is a closed source CMS and offers a free trial for any developer. Making the CMS close source will make it harder for people to hack it. Compared to the other CMS, EPiServer CMS releases updates frequently to ensure its users a safe website. Using EPiServer encryption will prevent security shortages and create a very secure environment [18, 22].

2.3.4 Popularity and trendiness

WordPress is one of the most used and popular CMS tools globally, with over 7 million created websites. The CMS tool is mainly used because it is open-source and used as a personal publishing platform [28]. Drupal has over 300000 websites created and allows its users to organize, manage and publish their content with a variety of customizations [29]. Umbraco has over 18000 websites created, and EPiServer has over 3000 created websites. EPiServer CMS is becoming a more trending CMS tool compared to the other CMS [30].

2.3.5 Developer experience

According to a study [18], users found Umbraco to be accessible and free to use with a helpful community. The CMS lacks documentation and is best suited for medium-sized projects. EPiServer was considered mature and robust with good community and support, but the CMS tool is expensive, and some minor technical solutions are not as desired. WordPress is considered a better option because the CMS tool is cheap, simple, well known, and has extensions available. The CMS tool has a complicated code base and has a long start up time to get the desired CMS. Drupal was considered adaptable, having a good structure, and handling both advanced and simple tasks. Drupal is hard to learn and requires plugins to be useful CMS.

2.3.6 Summary

After researching and analyzing the different CMS tools, we decided to use EPiServer CMS. WordPress is a popular CMS globally. This is considered a disadvantage because almost all developed websites have the same structure, which increases the risk of a breach or getting hacked. The CMS has many plugins developed by individual developers, which can cause conflicts with other plugins or the website to crash. Drupal takes a large amount of time to learn due to its advanced learning curve. The CMS is a good choice for larger projects, but with the risk of a slower content load. Drupal only provide their users with a limited number of free plugins. Umbraco is an acceptable choice for building intranets, but the CMS demands good experience and knowledge from developers due to the lack of documentation. Similar to Drupal, Umbraco only offers a limited number of free plugins. EPiServer also has an advanced learning curve but is well documented and provides good support for the usage of the CMS. The disadvantage with EPiServer is the cost, but the CMS offers 30 days of a free trial, which can be renewed. EPiServer offers its users various free plugins and is the best choice for creating an intranet that works for large systems and is adaptable for smaller ones. It is a secure system that provides the opportunity to develop a unique web application [18].

2.4 EPiServer CMS

EPiServer is a Swedish company, and their first version of the system was launched in 1997. The system was founded to handle and manage content on a website or similar systems. The company allows the customer to update and edit their websites using a web-based interface without any specific knowledge. Today, EPiServer has developed, and they count as one of the biggest suppliers within CMS systems [31].

EPiServer is a web publishing tool used to control internal and external web systems dynamically and is one of the options for developing dynamic web applications [22]. EPiServer CMS is a tool, part of a larger platform, and can be built up with several different products providing e-Commerce and personalization features. The CMS is built on the ASP.NET framework. Using the EPiServer CMS framework, the back-end programming is written with C# language.

EPiServer CMS contains multiple classes and repositories, and the content repository contains various content in the form of pages, blocks attached on pages, and media. The EPiServer CMS properties can have their metadata or treatment modified by using attributes such as AllowedTypes, Searchable, and CultureSpecific, which are located in the EPiServer.DataAnnotations namespace [32, 33]. The IContent interface properties are defined with the keyword "Virtual" because EPiServer's read operations use proxy classes when fetching data from the database; these proxy classes work with virtual properties [32]. The IContentRepository in EPiServer CMS is the primary API that defines repository methods for IContent objects. Through the repository, multiple operations such as, Create, Read, Update and Delete can be performed on content instances implementation using EPiServer.Core.IContent. The Content can be formed as BlockData, PageData, and BasicContent to create blocks, pages, and media content, as shown in figure 2 [34].

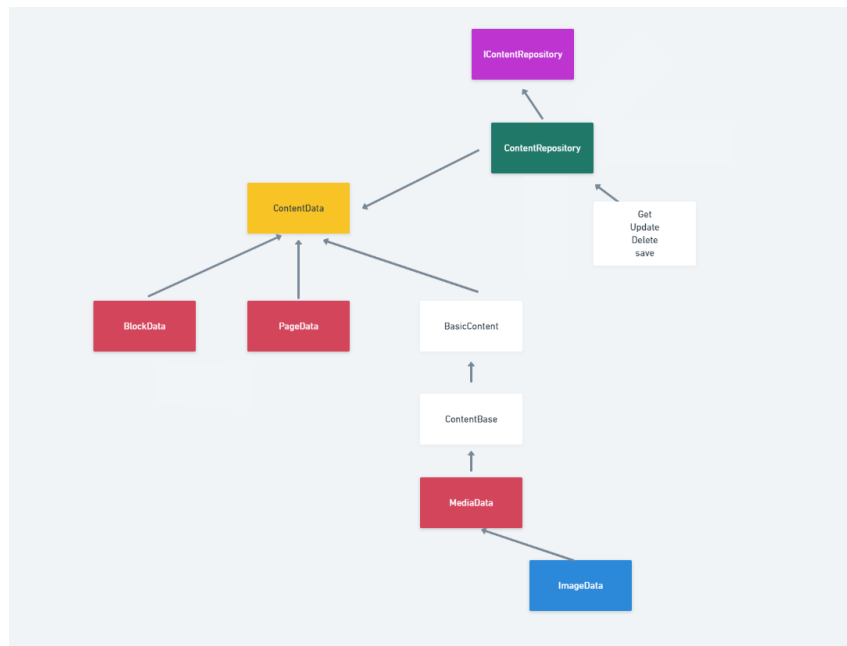


Figure 2. The content of IContentRepository [22].

In figure 2, the ContentRepository and its interface include the main functions such as Get, Update, Delete and Save. The ContentData is a class inherited from the ContentRepository class. BlockData, PageData, and BasicContent are classes inherited from the ContentData class. ImageData is an important class for adding images in the EPiServer Edit Mode. This class is inherited from the MediaData class, which is inherited from the BasicContent class. Page type class in EPiServer CMS is derived from the PageData class. A page type contains various properties which can be assigned by values in the EPiServer Edit Mode. The creation of Page types can be done either in code or from the admin view. Blocks are defined by inheriting from the BlockData class and are reusable content types. Blocks are used to add content to a page dynamically [33].

2.4.1 EPiServer Search & Navigation

EPiServer Search and Navigation, formerly known as EPiServer Find, is a Cloud-based search engine. The search engine is used for EPiServer CMS and EPiServer e-Commerce because it is natively created from EPiServer. Search and Navigation consists of a back-end software service and a front-end UI to build the site search functionality [35]. It is built using the Elastic search engine and provides various advanced features such as statistics, multi-language support, and searching from file content and content areas. EPiServer Search and Navigation offer the standard .NET client API via an instance of the IClient interface. It handles all interactions with the search engine, such as indexing, deleting, updating, and searching [33].

2.5 Tools and languages

2.5.1 ASP.Net

Active Server Pages (ASP) is a development framework for building web pages and was introduced in 1998 as Microsoft's first server-side scripting language. The ASP technology enables computer code to be executed by an internet server. ASP.NET was released in 2002 as a replacement to the classic ASP. The ASP.NET framework extends the .NET framework, a developer platform made up of programming languages, tools, and libraries to build various web applications. ASP.NET pages are generally written in the C# programming language. ASP.NET is one of the most widely used languages in web development [36]. It is an event-driven programming model, and the pages get access to classes of the

.NET framework. The web application can be coded in any language if it is compatible with the common language runtime. The pages created with ASP.Net are object-oriented and have rich APIs for handling database connections [37].

2.5.2 C#

C# is an object-oriented and type-safe programming language designed in 2000 by Microsoft. It enables a developer to create and build various applications that run in the .NET ecosystem. The C# language has its roots in the C language family [36]. C# has an application development to create applications such as intranets. The C# data types, including primitive types, inherit from a single root object type. The types share a set of standard operations. C# language allows dynamic allocation of objects. It supports generic methods and types, which increase type safety and performance [38].

2.5.3 JavaScript with React library

JavaScript (JS) is the primary scripting language used to develop and design dynamic web content. It is created by Brendan Eich, a Netscape programmer, in 1995. JS was originally named Mocha but quickly became known as LiveScript and JavaScript later [39]. Dynamic web content is the content that moves, refreshes, or changes on the user's screen without requiring any changes in the code, such as photo slideshow, animated graphics, and interactive forms [40]. For advanced web applications, many JavaScript libraries and frameworks have been developed. JavaScript contains various libraries depending on web applications such as Ember, Angular, Vue, and React [40].

The web application during this project has been developed using the React library. React is a JavaScript library created by Jordan Walke, a software engineer at Facebook, in 2011. In 2013 React became open-source, and in 2015, React Native, a library for mobile devices, became open-source. Facebook and Instagram are built on react, and the library is developed by Facebook and a community of individual users [41]. React is designed to simplify developers' jobs when interacting with UI development, and the view needs to be updated when the data change. The library divides the view into smaller components that can be composed to achieve more complex UIs. The components are built in JavaScript instead of templates, and this will enable an easy flow of data.

React has a programming concept called the virtual Document Object Model (DOM). Virtual DOM updates the browsers, and the virtual DOM concept represents the UI that is kept in the memory. The "real" DOM is synchronized via a library such as React DOM. The manual attribute and event handling gets removed and automates updating of the DOM [41].

2.5.4 HTML and CSS

The Hypertext Markup language (HTML) and Cascading Style Sheets (CSS) are two of the core technologies for the development of web pages. The HTML provides the structure of the web page, and CSS provides the layout. HTML was created by Tim Berners-Lee in 1993 and is used to publish the content on the World Wide Web (WWW). HTML uses tags to describe how the content should be formatted. The browser loads an HTML code, reads the tags, and builds the website. Every website should start with <html> and ends with </html> [42]. CSS is used to design the HTML code; it was developed by the World Wide Web Consortium in 1996. It allows the separation of content and presentation, including layout, colors, and fonts. The advantage of using CSS is that the developer can control the HTML page design, making it easier to create and update the overview of the website [42].

2.5.5 Visual Studio

Visual Studio is an Integrated Development Environment (IDE) and was first released in 1997 and was first known as Visual Studio 97. Visual Studio IDE was developed by Microsoft, and its purpose was to develop Graphic User Interface (GUI), console, Web applications, mobile apps, cloud, and web services [43]. Visual Studio is a non-language-specific IDE, and the program supports 36 different programming languages like C#, C++, Python, JavaScript, and many more. Depending on the development of games or web applications, Visual Studio is available for both Windows and Mac for different development purposes. During this project, Visual Studio Community was downloaded and used.

Visual Studio Community is a free version, and this edition consists of various limitations. Still, these limitations only apply to big organizations or non-enterprise organizations with more than five developers. The free version offers any individual developer to create their own free or paid apps like .NET applications and web applications [43].

2.5.6 GitHub

Git is a version control software that allows saving changes over time without overwriting previous versions [44]. GitHub is built on top of the git version control system, and it is a collaborative code hosting site and is one of the most important sources of software artifacts on the internet [45]. GitHub simplifies the developers' work to share and save the written code to other developers without the risk of losing the developed and updated files. Because of SHA compression in Git, the various actions are executed fast [44].

2.6 Types of intranets

Most companies, big or small, use an intranet to make a more efficient and productive environment between colleagues and employees. Depending on the company or organization, the development of the intranet will vary. In general, regardless of company or organization size, an intranet is not open to the public and contains the same primary functions. The development of the intranet is only for internal use and is only accessible by the employees. Intranet is secured behind the software, hardware, and firewalls to maintain the security within the company or organization, which keeps the infiltrators out. The implementation of the password function will ensure that authorized people can access the intranet. Some intranets need area restrictions to a particular group of employees working on a project together or members of a specific department [46]. In simple terms, big companies require a more developed intranet containing the functions required and needed by the company, while small companies require a simpler developed intranet. There are three general types of intranets:

- **Collaboration platform**

This type of intranet is a virtual workspace where projects and tools are gathered to facilitate communication and personal interaction in corporate projects. It will allow employees to get access to information and encourage collaboration in projects through the platform. There are different existing types of collaboration platforms that can be customized depending on the type of the company and its needs [47].

- **Internal web site**

This type of intranet is based on one-way publishing. The structure of the internal website is similar to a public website, but it is private and exists behind the firewall. There are two groups of people who interact with this kind of intranet: administrators and employees. Information is reviewed by administrators before it is published [48].

- **Distributed intranet**

This type of intranet is mostly created for larger organizations and companies. It is the place where the company gathers lots of its small applications such as a phone directory, an announcement system, and a digital library in one place. It is a combination of having both a collaboration platform and an internal website [48].

This project aims to create an intranet with EPiServer CMS and the required tools. The intranet will contain a simple structure and a UI with various primary functions. By only allowing administrators to publish the company news, allowing employees to create, share and delete events, and implementing a search functionality for faster information access, the best type of intranet to use is an internal website. The purpose of an intranet containing primary functions is to work for any kind of company and organization. A collaboration platform is a great option for creating an intranet, including a virtual workspace. To implement this type of intranet, it is important to know more about the business within the company. The implementation of this intranet depends on the individual company or organization and their needs; this means that the implementation of a collaboration platform intranet can only be directed at a specific company.

3 Method

In the following section, there is a review and information about various software that has been used during the project to create an intranet. Intranet only consists of software developments without the need to use hardware components.

3.1 Task Specification

To achieve a functional intranet with EPiServer CMS framework and JavaScript React library, the project needs to be divided into two major parts. Before starting with the coding process for each part, research had to be done about the specific frameworks and technologies. Because the subject of software web development is broad, the research was narrowed down due to the time. After researching, the back-end and front-end parts were specified with different instructions for easier development. A significant part of the project was to understand the usage and implementation of EPiServer CMS because the framework is the basis of the development of an intranet [32, 33]. The EPiServer documentation contains all the basic information for creating properties, attributes, and much more. Following the documentation allowed us to code the back-end part and implement the necessary functions to create an intranet compatible for companies. The front-end part is a React application that needs to be linked to the back-end part. To link the major parts, a study on how API and HTTP requests works had been done.

3.2 Starting up

An intranet is a software program with dynamic content built up by different programming languages (C#, PHP, JavaScript, etc.). For building a dynamic web application, it is essential to use a CMS tool. The primary task of using CMS is to make connections between the web server, database, and the browser. Another main task of using CMS is to create dynamic web pages on a request by gathering data from a file system or a database.

EPiServer CMS was chosen to build the structure of the intranet, which is the back-end part of this project, the EPiServer CMS framework is coded with the C# programming language. Using the Microsoft .NET framework opens possibilities to use a secure tool, which helps design, create, and distribute services and programs [49]. To develop an intranet using the EPiServer CMS features, an extension had to be downloaded from the Visual Studio marketplace shown in Figure 3.

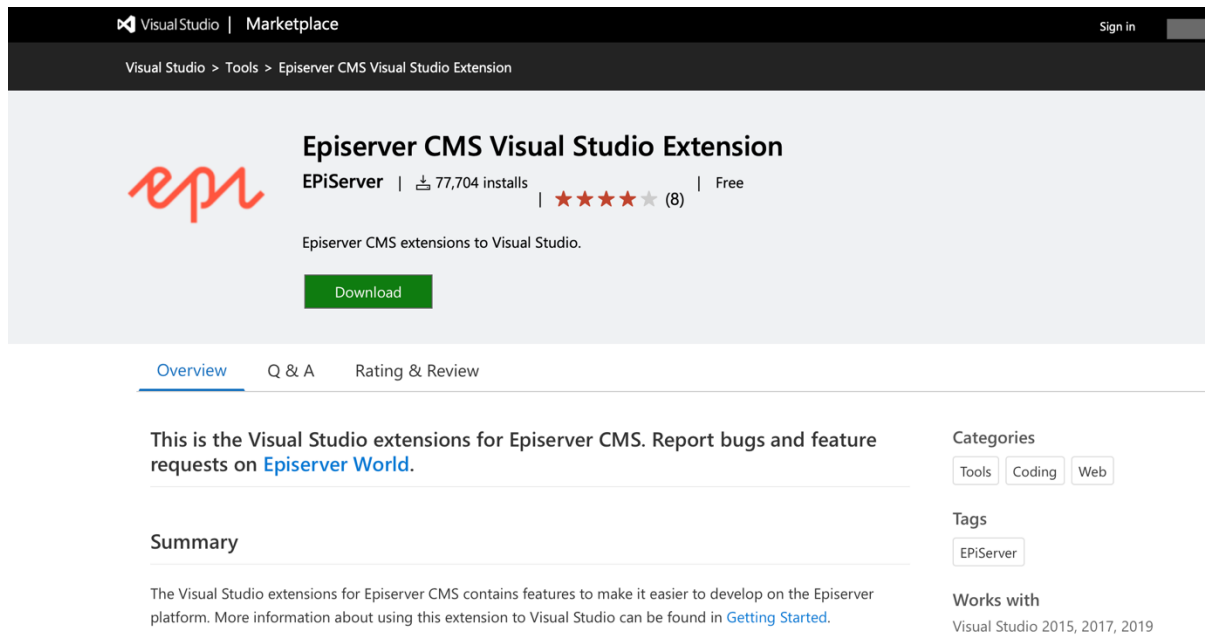


Figure 3. EPiServer CMS download at Visual Studio marketplace

Before starting with the project, structural planning was needed to finish the project in time. The back-end programming took most of the time. Creating a scheme and adding different deadlines to the project made the programming process more comfortable and knowledgeable. Creating logins was a must to become an administrator for EPiServer edit mode, allowing changes to the intranet content. Code sharing was a critical moment in building up the project. For being able to share the written code efficiently between the collaborators, GitHub was used.

3.3 Data access layer

3.3.1 Pages

To be able to create, add and edit the dynamic content from the UI, the structure of various pages has to be developed in the back-end. To achieve a functional intranet, various page types must be programmed, and each page type represents a specific website function and the content it includes. Each page type would consist of multiple properties depending on the functionality of the page.

The start point of the back-end programming was to adapt the API design pattern by dividing the code and creating multiple folders within the workspace using the Visual Studio community [50]; the main three folders are the following:

- Model: contains all the created page types.
- ApiController: contains all API classes needed to connect the front-end with the back-end.
- Services: contains all the needed functions divided into various classes.

The purpose of using the web API is to test the back-end functions and logic by sending HTTP requests before developing the front-end. Setting up the project in EPiServer Edit started by creating the first page of the website. Firstly, a new page type was developed specified just for the start page. The creation of the page type would be done programmatically using the PageData repository that is a feature from EPiServer CMS. The start page would work as the root of the program, and it would include multiple unique properties, as shown in figure 4. After developing the start page type, it was added in EPiServer edit, where it was created and linked as the root of the application. When the root was set, necessary page types were coded with specific properties and added to EPiServer Edit. Two important page types with identical structures with different purposes were created: Contact Us page type and About Us page

type. Both include an image property for inserting images and text field properties for adding headings and text, as shown in figure 5.

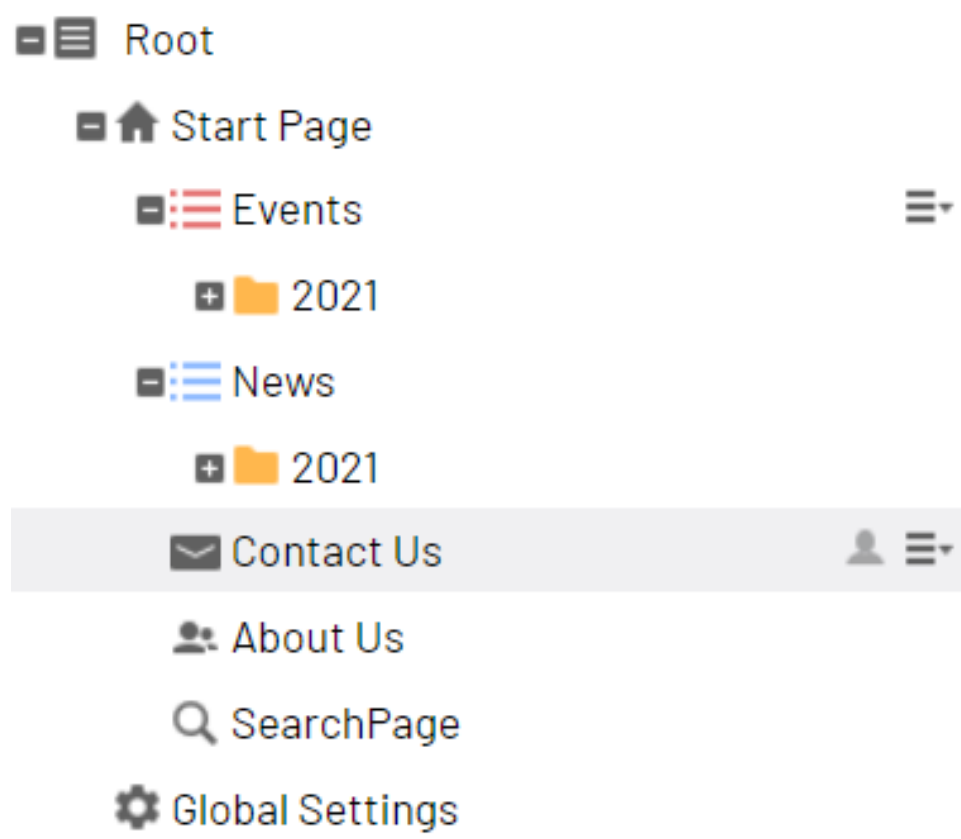


Figure 4. Start page represents the parent of the other pages in EPiServer edit mode

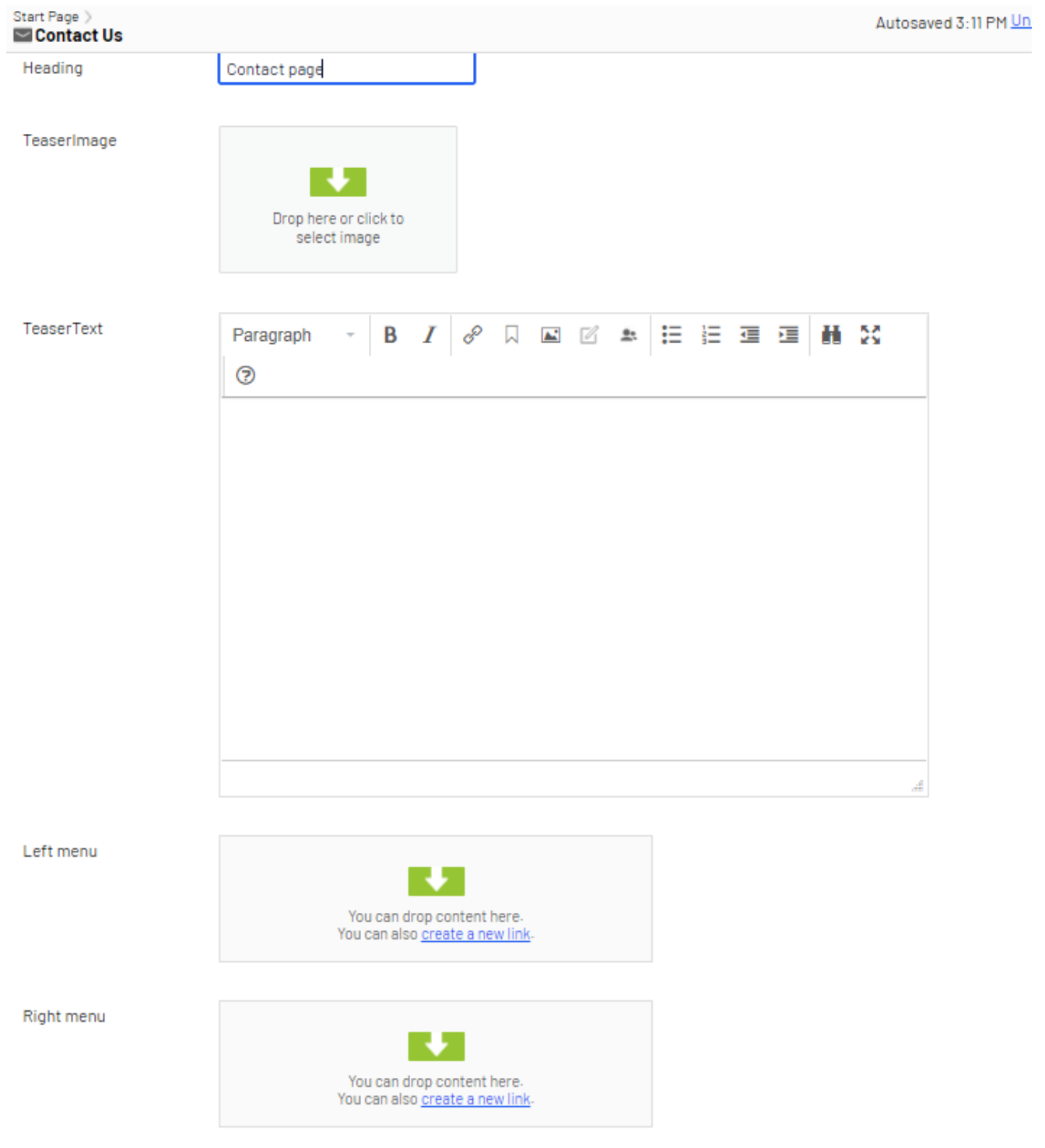


Figure 5. Contact us page containing various attributes in EPiServer edit mode, About us page contains identical properties.

The different page types were created by writing code in Visual Studio Community with C# to facilitate adding the pages in EPiServer edit mode. The number of created pages was due to primary functions for an intranet, and pages could easily be added or deleted depending on the needs.

3.3.2 News and events feed

A structured intranet needs various functions for the user to interact with, and this could be sharing news and events. To achieve adding these essential functionalities to the intranet, two new page types were coded with different unique properties. News and events page types share some properties, such as Filter tags, a customized property to filter news and events. The Filter Tags is needed because the intranet will be created for companies located in multiple countries and consists of different departments. The other purpose of Filter tags is to allow users to create specific groups for information sharing. Both page types include text fields and image properties for inserting images and adding title and text. News page type has a text property to add an article summary and event page type has a time property to select start and end times for the events. The event page type also has an address property and a meeting link that allows the user to write the address of the created event. To store news and events in a sorted and organized way, the created news and events would be stored in two lists and in different containers. Containers are page types developed specifically to store the news and event pages. The containers would work as folders, and they would be sorted after years and months. In this way, the searching process after specific news and events would be fast and efficient, as shown in figure 6.

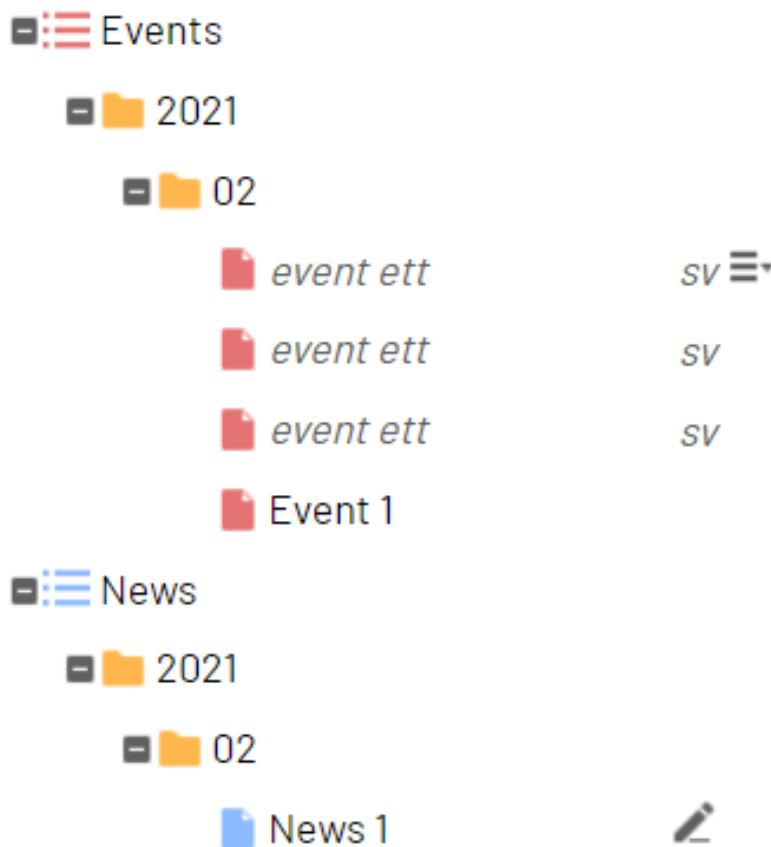


Figure 6. The events and news are stored in their designated folders marked with the specific date.

3.3.3 Tags and restrictions

As shown in figure 7 below, the added news and events can be created for a specific group of employees using the filter tags functionality. These customized properties are created using the SystemTabNames class from EPiServer.dataAbstraction, which is an EPiServer repository. These properties will get their values from the database, and because EPiServer handles the management of the database, new values can be added, edited, and deleted easily from EPiServer edit.

News and event page types share three filter tags:

- Country Tags: depends on where the company is located geographically.
- Department Tags: include the various departments within the company.
- Employee Level Tags: include the employee's level status.

The event page type consists of similar filter tags and an extra featured filter tag called Event Tags which contains different categories that describe the type of the event. This feature will help the user search for a specific type of event.

Content

Settings

Filter Tags

Country Tags

Sweden ×

Norway ×

+

Department Tags

IT ×

Security ×

+

Employee Level Tags

Expert ×

Lead ×

+

Article Title

Example of news using tags.

Figure 7. An example of filter tags used in a news page.

3.3.4 People List

Employees' profile pages are an important part of this project, and each one needs an individual profile. To make it easier for administrators to create an employee profile, a profile page type was structured and designed.

The profile page type includes various properties that make the profile structured and organized.

- Country: based on the employee's location.
- Department: based on the department that the employee works at.
- Employee Level: bases on employee's level status.
- Profile picture.
- Full name.
- Email.
- Mobile.
- Position: based on the employee's role within the company.

This functionality was added to allow users to acquire contact information about other users, this feature can facilitate the connection between the employees. This can facilitate the sharing between a specific group of people. Some profile changes need to be accepted by the administrators. The storing and sorting of the profiles are very similar to storing and sorting events and news. Every profile will be

stored in a list consisting of different containers that work like folders. These containers represent the country, department, and employee levels. Profiles with a shared place, department, or employee level will be stored in the same containers. This method will make searching for various profiles rapid.

3.3.5 Services and dependency injection

The foundation of this project is built upon the EPiServer CMS framework, and this framework includes many repository classes and functions. Repository classes are only used to connect the database for accessing and storing the information; these classes cannot have any other logic. To create classes that include various functions and logic, Service classes are used. Services are used to fetch data from a data source like a repository, handle the data depending on the logic and return the outcome to the caller. A service can collect data from multiple repositories to get the expected result [51]. In the Service folder, multiple service classes were created depending on the page type that might contain dynamic content. This is because every feature in the intranet needs developed logic to work. By dividing the logic into multiple services, the testing process would be uncomplicated, and the code would be reused in other services.

Dependency injection (DI) software design pattern is a technique that allows the application to inject objects to the classes that are dependent. The DI technique injects created objects from specific service classes into other service classes [52]. By using this technique, the services would interact with each other to achieve the required result. These services would connect to the database through EPiServer repositories, and the process depends on the included logic and send the result data to the database. With the DI technique, logic can be injected from one service to a different service, and this will avoid code cluttering.

When adding events, the user can enter multiple required inputs, which would be sent to the service to process, when the execution is done, the result would be sent to the database where it would be stored and to the presentation layer where it will be displayed for the user.

The services created in this project are:

- Event Service: contains logic to get an event, a list of events, update, delete and edit an event.
- News Service: contains logic to get news and a list of news.
- Tag Service: contains logic to get all types of filter tags.
- Container Service: contains logic to get a specific container and create a container automatically when news, events, and profiles are added.
- User Service: contains logic to create, update, delete and get user's profiles.

3.3.6 API Controller

API controller is a class that is created inside the controller folder under the project's root folder. All controller classes have to end with "Controller" and be derived from the `System.Web.Http.ApiController` class [53]. Web API controller handles the incoming HTTP requests and sends the response back to the caller. All public methods of the controller are called actions. To execute an action, the names must match with HTTP verbs such as GET, HEAD, POST, PUT, DELETE, CONNECT, and many more [54]. Based on the incoming URL and the HTTP request, the web API decides which API controller and action should be executed. For example, when a user wants to delete an event, a DELETE request is sent from the presentation layer to the `eventsApiController`. After finding the Delete function and matching it with the HTTP verb, the developed logic in the `eventService` is executed, and the result is sent to the database. When the result is received, the event is deleted from the database and the website. Figure 8 below shows a function from `eventApiController` that has an HTTP DELETE verb. This function calls the developed Delete function from the `eventService`.

```

[Route("Delete/{eventId}")]
[HttpDelete]
0 references
public void Delete(Guid eventId)
{
    _eventsService.Delete(eventId);
}

```

Figure 8. An example of an API method and its HTTP request.

The API controller classes created for this project are:

- Event API controller
- News API controller
- User API controller

All these API controllers contain various functions with different actions that match the HTTP requests. Every function calls a specific service to execute the logic, get the result, and return it back to the caller, in this case, the presentation layer.

3.4 Presentation layer

The presentation layer or the front-end is the part of the application that receives input from the user, and it is the interface that the user interacts with. It consists of the layout and the design of the application and the functions that let the user interact with the data in the back-end [55].

The start point was to create a React application in Visual Studio Code and divide it into multiple folders for building a structural workspace. React has proper documentation where main concepts are explained and provides developers with good knowledge to get started. React elements are plain objects and easy to create compared to browser DOM elements. Using React gives the possibility to implement third-party packages and frameworks [41]. Installing Node Package Manager allowed us to implement Material UI for creating a user-friendly UI.

3.4.1 Design

The design is one of the critical parts of this project, and it determines the user's experience when interacting with the intranet. The design of the intranet was created with the use of the Material UI React framework. Material UI allows the developers to create a smooth and straightforward interface for the usage of the application and obtain a high-quality user experience. The Material UI framework was downloaded to the project using the node.js command "npm install @material-ui/core". This package contains various customized React components that are used to reach a user-friendly design of the intranet [56]. The intranet front-end code was written with JavaScript React library and with the customized components from Material UI. The layout consists of various buttons, symbols, and images, and multiple pages:

- Sign in page will allow the user to fill the username and password to log in to the intranet.
- Main page consists of a dashboard containing the latest news and events.
- Menu that allows the user to navigate through the pages.

- News page displays the news list and allows the user to read the added news.
- Event page displays the events list and allows the user to create, delete and edit their own events and to read the events created by other users.
- People page allows the user to find all the registered profiles and their contact and job information.
- Account page displays the user information such as Name, Mobile Number, Email address, Position, Country, Department, and Employee level. This page allows users to change their name and mobile number if needed.
- Settings Page allows the user to change the profile's password.

To store the data before displaying it to the user or after receiving it as inputs from the user, a local store class was created. Having a local store made it possible to use React hooks. This will give the developer the possibility to use the React state and various features from React without the need to write a class. Adapting React hooks to the program allows the developers to extract logic from a component so it can be tested independently and reused. All the data were stored in the database, and when it reaches the front-end, it is saved in various properties within the local storage to be used later for executing some logic or be displayed to the user.

The code was divided into multiple folders inside the source folder of the React application. The folders are actions folder that includes all action classes, a component folder that contains the most important created components which are used in multiple classes, reducers folder where the local storage was defined, and a views folder including the view of the event page, news page, and the profile page

The logic of the application was coded inside the action classes, and the functions were defined and coded. The NewsActions class includes functions to get news, list of news. The EventActions class provides functions to get an event, an event list, add, edit, and delete an event. The ProfileActions class includes functions to get a profile, update user information and change the password. All the logic inside the action classes consists of sending HTTP requests with input data from the user through the API to the back-end, waiting for the data to process, getting an outcome, returning it to the caller, or displaying it to the user.

3.4.2 Requests

As mentioned before, the presentation layer and the data access layer communicate through sending HTTP requests and receiving responses via the API. In the front-end part, the HTTP requests are coded inside the action classes and define with JSON code [53]. The JSON code varies depending on the logic inside the functions in the back-end. If a process needs some parameters to execute and return a value, the parameters must be sent from the front-end in a JSON file form. The parameters inside the JSON code must have the same name as the parameters inside the functions in the back-end. Otherwise, the HTTP requests will execute with the wrong JSON values, and the response will include multiple errors. After receiving the returned values, they will be stored in the application's local storage and be passed to the view classes to be used inside other functions or to display it.

3.4.3 Routes

Routing gives the user the ability to navigate between different pages and parts of the web application. The navigation action can be done by clicking an element such as a button, a symbol, an image, or entering a URL through the browser. React uses the react-router-dom library, which includes several routers that are virtual and can be used to navigate within the application [41, 57]. The routes class was defined inside the react application and consists of multiple virtual customized paths. Each path navigates to a specific folder to render the code depending on the action. If the user is not logged in, all the entered paths will be redirected to the log in page.

3.5 Result analysis

The testing was critical to be certain that each major part worked. This was to make sure that no complication would occur when linking back-end and front-end parts together. The approach to test each part was by sending HTTP requests containing JSON code via the Postman application. Each request tests one or multiple functions in the code. If the request received the correct response, the function was developed correctly. Otherwise, the code needed to be debugged to solve the problem that had occurred in the code. To take extra precautions, the developed application was tested by trying all the interactive buttons, pictures, text and scrolling through the UI to make sure the program is fully functional and bug free.

4 Result

In the following section, the results of the developed and finalized intranet of the project are presented. The result consists of a functional intranet that contains six pages with dynamic content and multiple primary functions. The application's data access layer is built on the EPiServer CMS framework, and the presentation layer is built using the JavaScript React library. The complete intranet contains the following features:

Pages:

- Dashboard
- News List
- Event List
- People List
- Profile
- Sign in

Functionalities:

- Log in with a username and a password.
- Add events with images and information about the event.
- Editing and deleting personal events.
- Editing name and mobile number of the profile.
- Searching for specific news, event, and employees.
- Filter Tags to filter the content to a specific target.
- Changing the password of the account.

4.1 Pages

The intranet contains various pages. The first page is the sign in page where the user can log in to the application using their personal username and a password, as shown in figure 9.

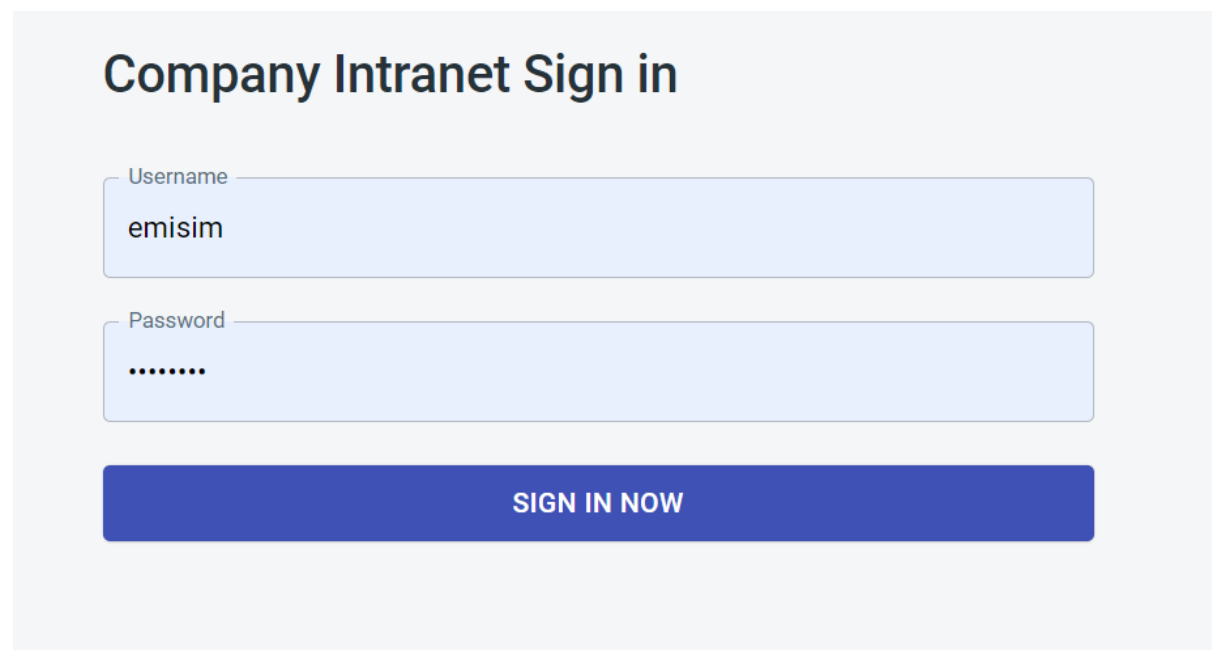


Figure 9. The sign in page to get access to the intranet.

The main page of the intranet is the dashboard page. In the dashboard, the user will see the main news, event, and a list of the latest news and events. Clicking on the news card or the event card allows the user to get access to them directly from the dashboard. The dashboard consists of different navigation options which the user can interact, as shown in figure 10.

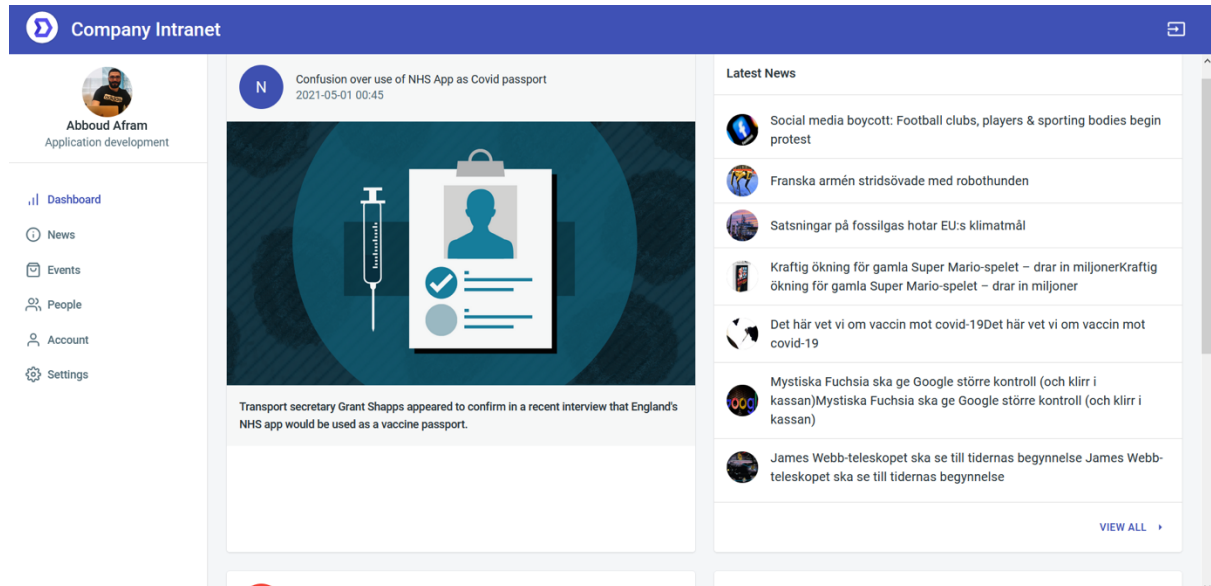


Figure 10. The user gets the following dashboard as the first page when signing into the website

The News List page contains all the news added to the intranet. The news is shaped like cards with a title, summary, and the created date. This page has a searching function. It searches for specific news by entering a keyword that matches with one or multiple news, as shown in figure 11.

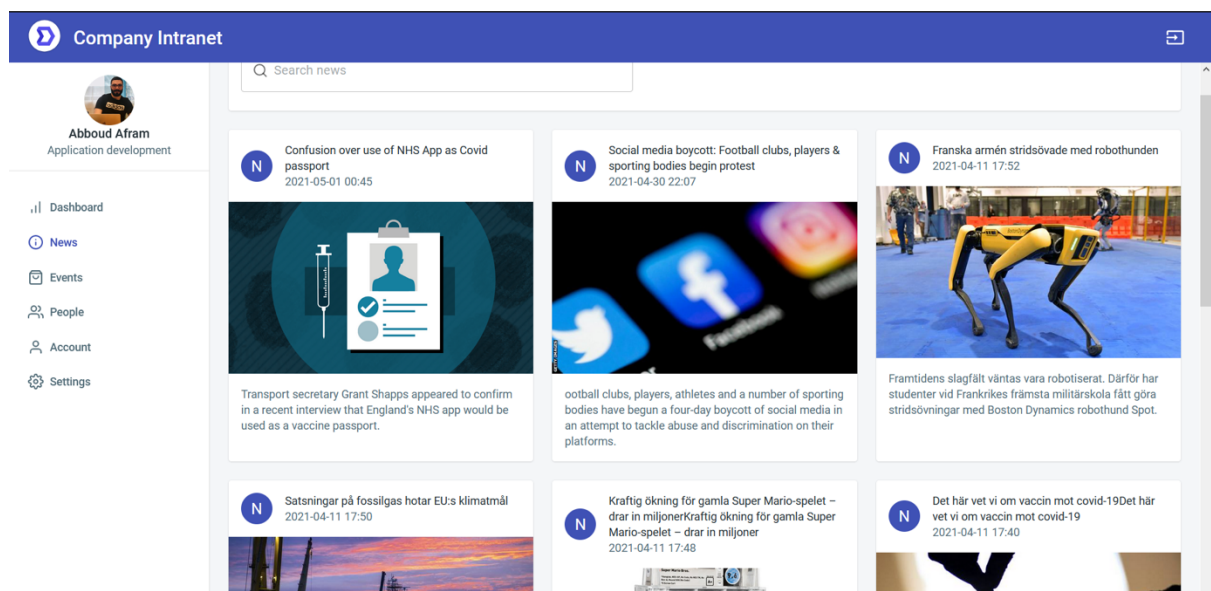


Figure 11. News cards for each news.

News Page opens when the user interacts by clicking on a news card. Inside the news page, the user can read the entire article, as shown in figure 12.

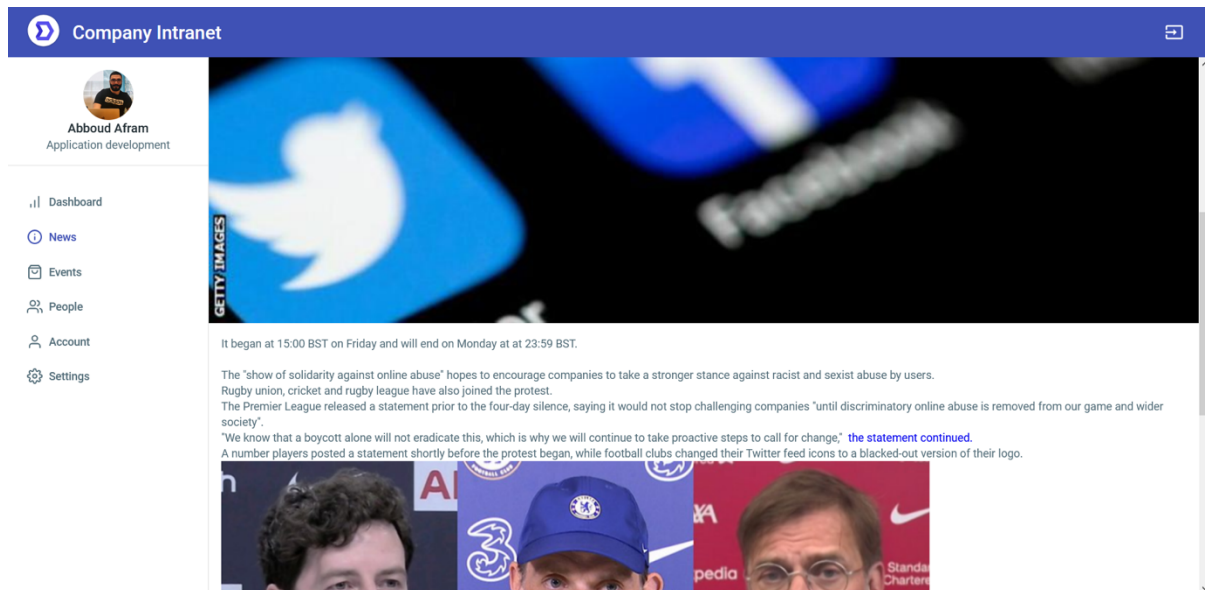


Figure 12. News page opens when the user interacts with a click and gets access to the article.

Event List page contains all the events added to the intranet. The events are shaped like a card with a title, Image, and the created date. This page has a search function, it searches for a specific event by entering a keyword that matches with one or multiple events. It also has an interaction button for adding new events, as shown in figure 13.

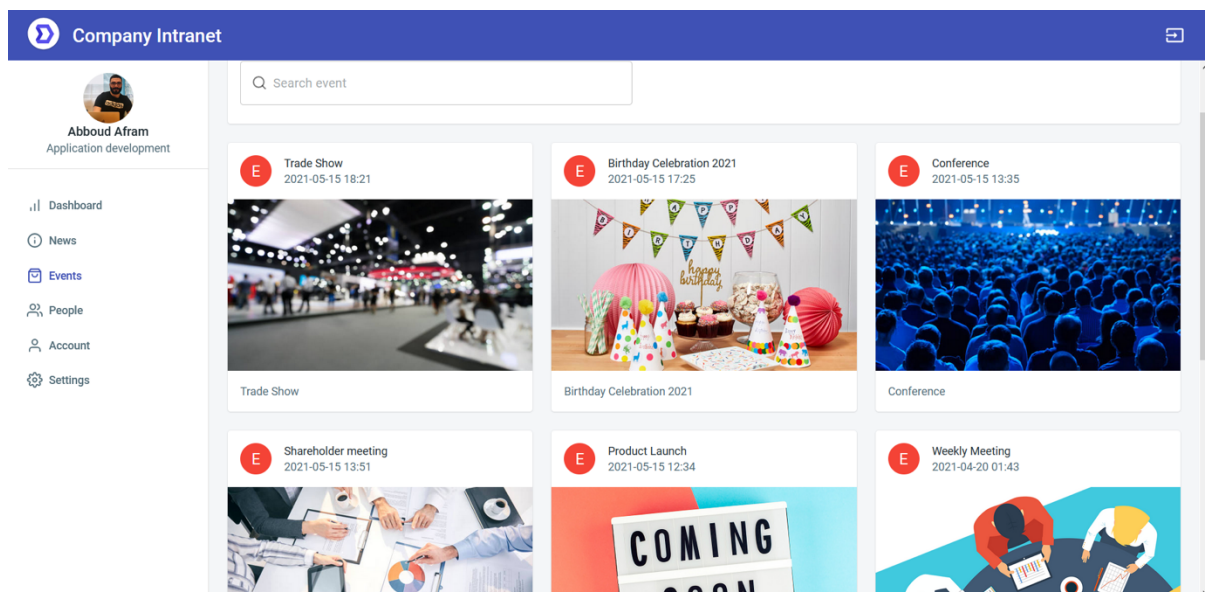


Figure 13. Event cards for each event, when the user interacts with a click the event page opens.

The Event Page opens when the user interacts by clicking on an event card, inside the event page the user can read the entire information about the event. The users will have the option to edit or delete their individual events, as shown in figure 14.

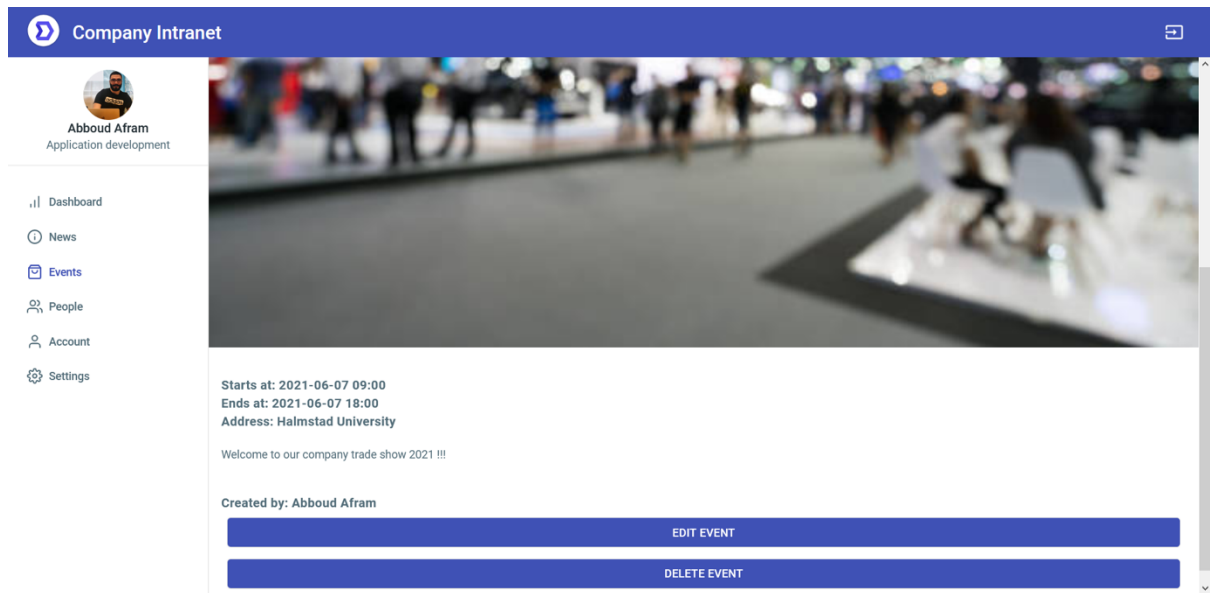


Figure 14. Event page opens when the user interacts with a click and get access to the article

The user has the option to add events by clicking on the add event button. After interacting with the button, the add event page will open. The add event page consists of filter tags, the name of the event, the address, description, setting start time, end time and date with the help of a calendar and insert an image if necessary, as shown in figure 15.

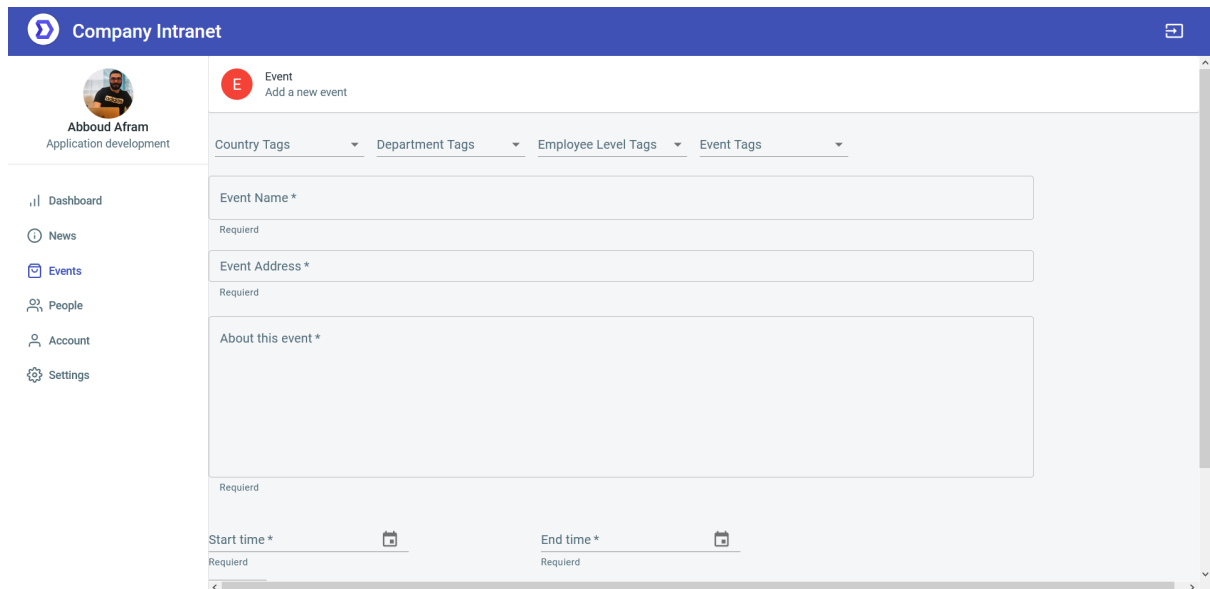


Figure 15. Adding options for creating and adding an event

People List page contains a list of all the employees currently working in the company. Each employee has a name, email, location, mobile number, and registration day. This page has a search function that allows users to search for a specific employee, as shown in figure 16.









| Company Intranet | | | | |
|--|------------------------------------|----------|------------|-------------------|
| <div>  <div> <div>Search people</div> </div> </div> | | | | |
| Name | Email | Location | Phone | Registration date |
|  Abboud Afram | abboud.afram@companyintranet.local | Halmstad | 0710000000 | 2021-05-15 |
|  Danial Sabet | danial.sabet@companyintranet.local | Halmstad | 0710000001 | 2021-03-07 |
|  FirstName SurName01 | emisim@companyintranet.local | Norway | 0700000001 | 2021-03-30 |
|  FirstName SurName02 | merbur@companyintranet.local | Denmark | 0700000002 | 2021-03-30 |
|  FirstName SurName03 | kwaseo@companyintranet.local | Finland | 0700000003 | 2021-03-30 |
|  FirstName SurName04 | anjkei@companyintranet.local | Halmstad | 0700000004 | 2021-03-30 |
|  FirstName SurName05 | adaden@companyintranet.local | Malmö | 0700000005 | 2021-03-30 |

Figure 16. People List page containing all the employees in the company.

Profile page contains information about the individual profiles such as name, email, mobile number, country, department, positions in the company, and employee status. The information is related to the company, and only the name and mobile number can be changed, as shown in figure 17.



| Company Intranet | |
|---|---|
| <div>  <div> <div>Dashboard</div> <div>News</div> <div>Events</div> <div>People</div> <div>Account</div> <div>Settings</div> </div> </div> | <div>  <div> <div>Abboud Afram</div> <div>Application development</div> </div> </div> <div> <div> <div>Profile</div> <div>The information can be edited</div> </div> <div> <div> <div>Full name *</div> <div>Abboud Afram</div> </div> <div> <div>Mobile Number</div> <div>0710000000</div> </div> </div> <div>Please specify the Full name</div> <div> <div> <div>Email Address *</div> <div>abboud.afram@companyintranet.local</div> </div> <div> <div>Position *</div> <div>Application development</div> </div> </div> <div> <div> <div>Country / State *</div> <div>Halmstad</div> </div> <div> <div>Department *</div> <div>Development</div> </div> </div> <div> <div> <div>Employee Level *</div> <div>Lead</div> </div> </div> <div>SAVE DETAILS</div> </div> |

Figure 17. Profile page containing profile information.

Settings Page is where the user can change the profile password, as shown in figure 18.

The screenshot displays the 'Company Intranet' interface. At the top, a blue header bar contains the 'Company Intranet' logo and a user profile icon. Below the header, a sidebar on the left lists navigation options: Dashboard, News, Events, People, Account, and Settings (highlighted). The main content area is titled 'Password' and 'Update password'. It features three input fields: 'Old Password *' (filled with dots), 'New Password *', and 'Confirm new password *'. A blue 'UPDATE' button is located at the bottom right of the form.

Figure 18. Change password functionality.

The fully developed intranet contains all the primary functions compatible for any company of any size to use. The plan was to add and develop more functions to target a specific tech company containing a secure option to chat and send documents with specific employee roles. Due to the time of this bachelor thesis, the non-primary functions were not added, but the intranet can continue to develop to target a specific company. The current intranet targets big companies and contains different filter tags like Nordic countries, departments, employee levels, and employee roles as a demo of an Internal Web Site and can easily be changed and adapted for any company's needs. The created intranet can be further developed, and more functions can be implemented to make it a useful tool for a specific company.

5 Discussion

We believe that the achieved results are in good agreement with the defined objectives of the project, as a functional and complete intranet was managed to be developed along with acquiring a lot of knowledge about software development. The time plan and schedule were followed correctly, and the project goals were reached with some modifications and exceptions. Due to the lack of time, some of the non-primary functions were not implemented to the created intranet, this was mainly the chat function which would allow the users to transfer files securely, chat with each other and target a specific company. This function was too complicated and time-consuming to implement, that we decided to devote more time to refine the UI and to add primary functions.

The finalized intranet is not as advanced compared to some of the intranets out on the market. Depending on company or organization, the implemented functions in our intranet differ by a large margin due to multiple factors such as the size of the company, the number of employees, and the branch. Compared to some web applications, in this project, both the back-end and front-end part was coded and linked while many other intranet parts are either created in teams, or one part is created in groups [42]. The finalized intranet contains the necessary functions and is still an ongoing project where implementation of more functions is possible to be suitable for a specific company and its requirements.

Testing the finalized intranet with different methods such as debugging and checking all functions by sending and receiving JSON data through the postman program allowed us to check the web application for any bugs. The project was not bug free due to the free version of the framework, and because of no economic support, the free version of EPiServer CMS did not allow some implemented functions to execute fast. When creating an event, the website needs to be updated a couple of times before showing the created event. This is due to the free version of EPiServer Search and Navigation and cannot be avoided. This is the major problem with the intranet, otherwise the web application runs smoothly with JavaScript React library, and no other bugs occur while running the program.

5.1 Company implementation of our intranet

The main purpose of the development of an intranet for larger companies is to create a good work environment and allow the knowledge within the firm to flow. Therefore, some of the functions implemented have a restriction, and only administrations of the company are allowed to change it. Depending on which company you are employed at, you are only allowed to change name, password, and phone number, not your picture, department, company mail, or status. Because all the information is company related and needs to be requested from administrators to change. The restrictions do also imply on company news. To be able to add news, a trusted source is necessary, and not everyone is allowed to post and share news on an internal website. The news functions are only to keep the employees updated about the company and the outside world. Event function is available for any employee to use, and they have the option to filter for specific countries, departments, or employee status, they also have the option to edit and delete their own events. The event page has an implemented function that allows the user to choose the date through a calendar and write the designated time. The main purpose of the event function is to allow the employees in the company to share their knowledge with colleagues through meetings, conferences, and lectures. There is also a search feature that allows employees to search for specific news, events, or their colleagues and their information by searching for the name or location. The intranet consists of an easy navigated UI with different buttons, symbols, and images to interact with. This allows multiple options and various interactions for the users when they want to load more news and events to read or simply to scroll through the profiles.

5.1.1 Social impact

Reviewing the ethic of this project is important, one of the reasons to implement an intranet to the company is to allow employees to communicate, share knowledge and expertise. Employees who have been isolated from the organization can now interact through the intranet. Allowing people to connect with the organization remotely and sharing information could solve some of the dissensus felt in the information [58]. But this may not always be the case, some of the employees may fear sharing their knowledge. The reason could be that employees do not want to become redundant simply because they do not want to share their expertise and/or avoid being embarrassed. Trust within the company is key for the successful use of the intranet. A study shows that cooperative behavior may emerge when trust exists, which leads to a positive association between trust and the extent of resources exchanged within units of a firm. Thus, companies or organizations that offer their employees a caring and trusting environment would be more likely to implement intranets successfully [59].

5.1.2 Safety and security

As any developed program, there will always be a vulnerability. The number of websites and services attacked has increased in recent years. At the same time, more and more companies have expanded on the internet and are discovering opportunities to market their company and ideas. Many companies do not priorities security which leads to web and application attacks. EPiServer CMS is a powerful development tool, and in general, EPiServer is a very secure platform [18]. The framework contains very few security flaws, and the deficiencies that exist are small and marginal. If an encrypted connection is implemented according to EPiServer recommendation, the system will be very secure [60].

6 Conclusion

With the result of this project, the goal was reached, a functional intranet demo adaptable and extendable for any company was reached successfully. Our intranet contains primary functions such as individual profiles with username and password, search function, a dashboard displaying latest news and events, event sharing with specific colleagues, and news sharing. Due to the time limit, some of the functionalities that were supposed to be added for users to interact with were not created. The intranet was supposed to contain a pinboard and file exchange function. To create these functions, a chat feature needed to be implemented, and according to us, this feature would be time-consuming and unnecessary.

We believe that the complete intranet facilitates and improves the work environment for the employees. Company news and events can easily be shared within the web application, as well as finding contact information about colleagues. The project contained a broad learning phase, the development of a web application and the programming languages were new but familiar. Based on previous and acquired knowledge, the coding part was not time consuming, but the code structure of new programming languages and all the EPiServer functions placed inside different repositories were the most time-consuming. The EPiServer CMS tool has been proven to be an excellent tool for the development of web applications. The framework takes time to learn and understand but has an easy-to-understand structure.

To develop and create an advanced intranet, usually, a team is needed. Where the team members are educated on the specific task depending on back-end or front-end programming. This will facilitate their work, collaboration, and a fully developed product. This project was instructive and truly fascinating, the web application software subject belongs to computer science, and it is interesting how easy it is to develop your own web application with the variety of programs out on the market. Acquiring new knowledge about different tools and languages is a great motivation to continue the bachelor thesis to develop a more advanced intranet containing many different features.

7 References

1. Leung, Hareton KN. "Quality metrics for intranet applications." *Information & Management* 38. (2001): 137-152.
2. Daya, Bhavya. "Network security: History, importance, and future." *University of Florida Department of Electrical and Computer Engineering* 4 (2013).
3. Subramaniam, Anand, and Hashem M. Ebrahimi. "Secure intranet access." U.S. Patent No. 6,081,900. 27 Jun. 2000
4. Stenmark, Dick. "Designing the new intranet." *rapport nr.: Gothenburg studies in Informatics* 21 (2002).
5. Lee, Sangjae, and Byung Gon Kim. "Factors affecting the usage of intranet: A confirmatory study." *Computers in Human Behavior* 25.1 (2009): 191-201.
6. Scott, Judy E. "Organizational knowledge and the intranet." *Decision Support Systems* 23.1 (1998): 3-17.
7. Edenius, Mats, and Janet Borgerson. "To manage knowledge by intranet." *Journal of knowledge management* (2003).
8. Carlsson, Peter. "Intranet: from the user's perspective. Department of Computer Science." (1997).
9. Oppliger, Rolf. *Internet and intranet security*. Artech House, 2001.
10. Ignatius, Per. "Security testing of web applications and the CMS platform EPiServer." (2011).
11. Banks, Alex, and Eve Porcello. *Learning React: functional web development with React and Redux*. "O'Reilly Media, Inc.", 2017.
12. Techopedia. "What is Web Development." *technopia.com*, 31 August 2020.
13. Wales, Michael. "3 Web Devs Careers Decoded: Front-End vs Back-End vs Full Stack." *udacity.com*, 8 December 2020.
14. Han, Yan. "Digital content management: the search for a content management system." *Library Hi Tech* (2004).
15. Irlbeck, Sonja, and Joanne Mowat. "Learning content management system (LCMS)." *Learning Objects: standards, metadata, repositories, and LCMS* (2007): 157-184. 15
16. Sindekar, Vaishali A., and Anup Kumar S. Pandey. "Comparison Between Traditional Web Design and CMS." *SAMRIDDHI: A Journal of Physical Sciences, Engineering and Technology* 13.SUP 1 (2021): 49-52.
17. Jönsson, Emil, and Andreas Nilsson. "Development and security evaluation of a flexible authentication solution." (2021).

18. Albrecht, Stefan, and Morgan Axelsson. "Umbraco, EPiServer, WordPress, Drupal and self-developed CMS-Developer opinion." (2016).
19. Bhandari, Mahesh. "Comparison of Wordpress, Joomla and Drupal." (2020).
20. Karthikeyan, S. "Advantages And Disadvantages Of Using Drupal CMS For Your Website." digisecrets.com (2021).
21. Rolina, Daria. "EPiServer VS Umbraco." ukad-group.com, 15 January 2019.
22. Jones, Jon. "Beginners's Guide: What is Episerver?". Jondjones.com, 3 August 2016.
23. Patel, Savan K., V. R. Rathod, and Jigna B. Prajapati. "Performance analysis of content management systems-joomla, drupal and wordpress." *International Journal of Computer Applications* 21.4 (2011): 39-43.
24. Sivaran, Jani. "Disadvantages of Using More Plugins in WordPress Site." medium.com, 12 July 2019.
25. Killpack, Kaili. "8 Reasons Not to Use WordPress." business.com, 15 April 2019.
26. Daniel. "Weighing Drupal's pros and cons: is it the right CMS for you?" eurodns.com, 12 April 2021.
27. Ryan. "Choosing an open source CMS: Umbraco vs WordPress." screenmedia.co.uk, 25 April 2019.
28. Similar Tech. "EPiServer VS WordPress". Similartech.com. (2021).
29. Similar Tech. "EPiServer VS Drupal". Similartech.com. (2021).
30. Similar Tech. "EPiServer VS Umbraco". Similartech.com. (2021).
31. J Öfverstedt, Linn. "Why go headless—a comperative study between traditional CMS and the emerging headless trend." (2018).
32. Riisalo, Henri. "Site search solutions in Episerver CMS." (2019).
33. Developer Community. "Property attributes". world.episerver.com, 19 March 2019.
34. Developer Community. "IContentLoader and IContentRepository". world.episerver.com, 26 October 2019.
35. Developer Community. "Architecture and languages". world.episerver.com, 05 November 2020.
36. Microsoft. "What is ASP.NET". dotnet.microsoft.com, 2021.
37. Mishra, Atul. "Critical Comparison Of PHP And ASP .NET For Web Development." *International Journal of Scientific & Technology Research* 3.7 (2014): 331-333.
38. Wagner, Bill, and David Coulter. "A tour of the C# language" docs.microsoft.com, 28 January 2021.
39. T.J DeGroat, "The History of JavaScript" springboard.com, 19 August 2019.
40. Duvander, Jacob, and Oliver Romhagen. "What affects the choice of a JavaScript framework: Interviews with developers." (2019).
41. Saks, Elar. "JavaScript Frameworks: Angular vs React vs Vue." (2019).

42. Oskarsson, Carl, and Jonathan Hedman. "Analysis and implementation of EPiServer." (2007).
43. Lee, Terry, and Gordon Hogenson. "Welcome to the Visual Studio IDE." docs.microsoft.com, 19 March 2019.
44. Naour, Lucie. "Git guide" github.com, 24 February 2021.
45. Kalliamvakou, Eirini, et al. "The promises and perils of mining github." *Proceedings of the 11th working conference on mining software repositories*. 2014.
46. Leung, H. K. (2001). Quality metrics for intranet applications. *Information & Management*, 38(3), 137-152.
47. Humanperf. "The benefits of a collaborative platform to the company." humanperf.com, 31 May 2018.
48. Barker. Deane. "The three types of Intranets". deanbarker.net, 11 October 2006.
49. McCune, J. C. (1996). The Intranet: Beyond email. *Management review*, 85(11), 23.
50. Microsoft. "Create web APIs with ASP.NET Core" docs.microsoft.com, 2020.
51. Kocsis, Zoltan A., and Jerry Swan. "Dependency injection for programming by optimization." *arXiv preprint arXiv:1707.04016* (2017).
52. Passos, Erick B., et al. "Smart composition of game objects using dependency injection." *Computers in Entertainment (CIE) 7.4* (2010): 1-15.
53. Chen, Xianjun, et al. "Restful API architecture based on laravel framework." *Journal of Physics: Conference Series*. Vol. 910. No. 1. IOP Publishing, 2017.
54. MDNWEBDOCS. "HTTP requests methods." developer.mozilla.org.
55. Nestler, Tobias, Lars Dannecker, and Andreas Pursche. "User-centric composition of service front-ends at the presentation layer." *Service-Oriented Computing. ICSOC/ServiceWave 2009 Workshops*. Springer, Berlin, Heidelberg, 2009.
56. Boduch, Adam. *React Material-UI Cookbook: Build captivating user experiences using React and Material-UI*. Packt Publishing Ltd, 2019.
57. Atto, Edmond. "Understanding the Fundamentals of Routing in React." Medium.com (17 Jan 2019).
58. Bennett, Elisabeth Eleanor. *How organizational culture and change are embedded in an organization's intranet*. Diss. University of Georgia, 2006.
59. Ruppel, Cynthia P., and Susan J. Harrington. "Sharing knowledge through intranets: A study of organizational culture and intranet implementation." *IEEE transactions on professional communication* 44.1 (2001): 37-52
60. Ignatius, Per. "Security testing of web applications and the CMS platform EPiServer." (2011).

Danial Sarab Fard Sabet

Abboud Afram



PO Box 823, SE-301 18 Halmstad
Phone: +35 46 16 71 00
E-mail: registrator@hh.se
www.hh.se