Application of Tailwind for Responsive Design in Enterprise Resource Planning (ERP) Website PT. Uniteda Arkato

PT. Uniteda Arkato
Jl. Pinang Ranti II No.44, Pinang Ranti, Kec. Makasar, Kota Jakarta Timur, Daerah Khusus Ibukota Jakarta 13560
Periode: 15 Juli 2024 - 15 Oktober 2024

Oleh:

Fauzan Ahmad Faisal   5025211067

Pembimbing Jurusan
Dr. Anny Yuniarti, S.Kom., M.Comp.Sc.
Pembimbing Lapangan
Muhammad Agus Arif Rahmatullah

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2024

Application of Tailwind for Responsive Design in Enterprise Resource Planning (ERP) Website PT. Uniteda Arkato

PT. Uniteda Arkato
Jl. Pinang Ranti II No.44, Pinang Ranti, Kec. Makasar, Kota Jakarta Timur, Daerah Khusus Ibukota Jakarta 13560
Periode: 15 Juli 2024 – 15 Oktober 2024

Oleh:

Fauzan Ahmad Faisal   5025211067

Pembimbing Jurusan
Dr. Anny Yuniarti, S.Kom., M.Comp.Sc.

Pembimbing Lapangan
Muhammad Agus Arif Rahmatullah

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2024

[This page is purposefully left blank]

TABLE OF CONTENTS

[This page is purposefully left blank]

FIGURE LIST

[This page is purposefully left blank]

TABLE LIST

[This page is purposefully left blank]

LEMBAR PENGESAHAN
KERJA PRAKTIK

Application of Tailwind for Responsive Design in Enterprise Resource Planning (ERP) Website PT. Uniteda Arkato

Oleh:

Fauzan Ahmad Faisal   5025211067

Disetujui oleh Pembimbing Kerja Praktik:

1. Dr. Anny Yuniarti S.Kom.,
    M.Comp.Sc.
    NIP. 198106222005012002

(Pembimbing Departemen)

2. Muhammad Agus Arif
    Rahmatullah

(Pembimbing Lapangan)

[This page is purposefully left blank]

Application of Tailwind for Responsive Design in Enterprise Resource Planning (ERP) Website PT. Uniteda Arkato

Student Name	: Fauzan Ahmad Faisal
NRP	: 5025211067
Department	: Informatics FTEIC-ITS
Department Supervisor	: Dr. Anny Yuniarti S.Kom., M.Comp.Sc.
Field Supervisor	: M. Agus Arif Rahmatullah

ABSTRACT

PT. Uniteda Arkato is a company focused on providing heavy equipment rental and contractor services based in Jakarta, Indonesia. In this practical work, the project I worked on is part of the development of an Enterprise Resource Planning (ERP) system that Arkato is developing. This system is designed to help Arkato employees manage their work online and more efficiently.

The system itself is a web application built using the Laravel framework. It already has many features, most of which are tailored to the different departments within Arkato. In this project, my task was to work on the interface to make the web application more responsive and user-friendly in terms of design.

Key Words: ERP, Website, Laravel, Framework, Tailwind

[This page is purposefully left blank]

FOREWORDS

The author would like to express his gratitude to Allah SWT for His guidance and blessing. It is because of His guidance and blessing that author can finish one of the requirements as a student in the department of Informatics ITS which is Practical Work with the title: Application of Tailwind for Responsive Design in Enterprise Resource Management Website PT. Uniteda Arkato.

Acknowledging the many shortcomings and mistakes that might appear while writing this report, the author still wishes that this report can widen the reader's knowledge and might prove sufficient as a future reference.

On this report, author would also like to express his thanks and gratitude towards the people that have helped in assembling this practical work report, be it directly or not. These people include:

1. Both of author's parents.
2. Dr. Anny Yuniarti S.Kom., M.Comp.Sc. as a supervisor for this practical work.
3. Mr. M. Agus Arif Rahamatullah as a field supervisor while commencing the practical work.
4. Friends and family that has supported author while doing practical work.

Surabaya, 12 November 2024
Fauzan Ahmad Faisal

[This page is purposefully left blank]

CHAPTER I
INTRODUCTION

1.1. Background

Resource management is one of the backbones of enterprising. Without proper resource management a company could fail or even go bankrupt. In this current state of the world, companies have taken their time to develop for themselves a resource management system that integrated all of their offices, be it main office or its branch offices. This can be achieved by using a website application that is connected to the internet and acts as an Enterprise Resource Planning (ERP).

This is what PT. Uniteda Arkato has been planning for quite some time. The company has a lot of branch offices and project sites in other part of the country. Therefore, they needed an integrated system such as ERP to manage all the resources that they have throughout the company. This is the reason why they are building a website application.

However, one of the problems that the developers has encountered when building the ERP website was the responsivity of the user interface (UI). The website was built originally for desktop use. However, some of the employees who doesn't always have the time to carry a laptop uses it on their phone instead. Since the UI designs are originally built for desktop use, when opened in a phone browser the elements became so jumbled all over and overall unusable. To tackle this problem the developer then decided to revise the tailwind code for the front-end part of the website. In this practical work, the author will be focusing on that particular task.

1.2. Goals

The goal in this practical work is to help PT. Uniteda Arkato in developing responsive design in their ERP system.

1.3. Benefits

The benefits that are gained from this practical work are PT. Uniteda Arkato will have a more responsive UI in their ERP website, helping the employees that use the website through their smartphone. Other than that, this practical work also benefits the author in learning more about tailwind.

1.4. Problem Statement

The problem statements from this practical work are:
1. How to achieve responsive design for PT. Uniteda Arkato ERP Website.
2. How effective is the responsive design after tailwind is applied.

1.5. Location and Time of Practical Work

This practical work is commenced on site from 15th of July until 6th of September. The location itself is the head office of PT. Uniteda Arkato.

From 9th September 2024 until 15th of October 2024 this practical work is done remotely. The reason is because the author has to attend to college classes in Surabaya and the head office of PT. Uniteda Arkato is located in Jakarta.

1.6. Practical Work Methods

The methods in writing this practical work are:

1.6.1. Problem Formulation

Since this is an ongoing project, author is first introduced to the ERP system and how the flow goes. Then, author is briefed about the UI problem that the developer team are facing. After the briefing author is shown the web pages with UI that needed to be responsive.

1.6.2. Literature Review

After having known which web pages needed UI revision, author is then told to made modification on the UI using tailwind. Author then reviews and learn more about tailwind and how to effectively apply it to the ERP system. Additionally, author look more on Laravel as it is the framework of the website.

1.6.3. System Development Analysis

The development team used the MVC (Model - View - Controller) architecture to organize and manage the system effectively.

1.6.4. System Implementation

The implementation involved deploying the revised Tailwind for responsive design and integrating backend components using Laravel.

1.6.5. Evaluation

The system was evaluated by the field supervisor. The supervisor used usability testing, performance testing, and feedback collection to ensure it met the objectives.

1.6.6. Conclusion and Suggestion

The system now has a more responsive UI on some pages. The UI has been evaluated to meet the company's needs and user's behaviour.

1.7. Report Systematics

1.7.1. Chapter I Introduction

This chapter includes the background, goals, benefits, problem statement, location and time of practical work, methodology, and report systematics.

1.7.2. Chapter II Company Profile

This chapter provides an overview of PT. Uniteda Arkato, including the company profile and its location.

1.7.3. Chapter III Literature Review

This chapter covers the theoretical foundation of the technologies used to complete the practical work project.

1.7.4. Chapter IV System Analysis

This chapter discusses the system analysis phase of the application development process during the practical work.

1.7.5. Chapter V System Implementation

This chapter describes the stages undertaken during the implementation process of the application.

1.7.6. Chapter VI Evaluation

This chapter presents the results of testing and evaluation of the developed application during the practical work.

1.7.7. Chapter VII Conclusion and Suggestion

This chapter provides the conclusions and suggestions derived from the practical work process.

CHAPTER II
COMPANY PROFILE

2.1.   PT. Uniteda Arkato Profile

PT. Uniteda Arkato is a company responsible for managing and overseeing various projects and operations across multiple branch offices and project sites throughout the country. It aims to efficiently handle its resources through integrated systems such as ERP, ensuring smooth and effective administration. The company's core functions in this particular project include implementing strategic resource management, coordinating between different offices, and executing tasks in line with corporate objectives and goals.

2.2.   Company Structure

Figure 2.1 shows the current company structure of PT. Uniteda Arkato.
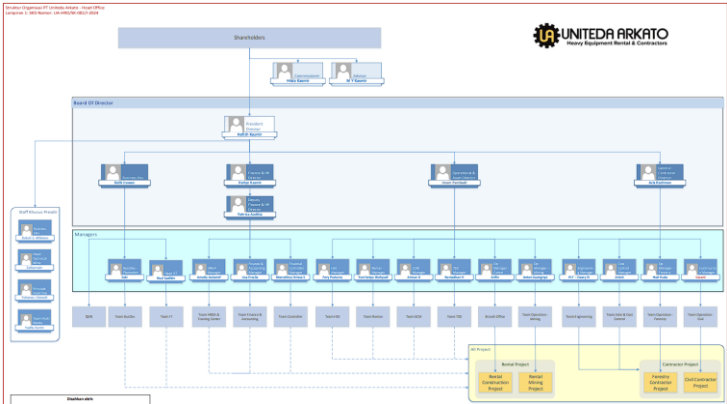
Figure 2.1 Organization Structure of PT. Uniteda Arkato

The company structure consists of 6 sections. These sections are:
1. Shareholders
2. Board of Director
3. Managers
4. President's Executive Staffs
5. Head Office Employees
6. Project Site's Employees

In this practical work author is assigned to the IT team. The IT team itself is managed by Mr. Rizal Sadikin and handles the IT support of the head office and other branches.

2.3. Location

Jl. Pinang Ranti II No.44, Pinang Ranti, Kec. Makasar, Kota Jakarta Timur, Daerah Khusus Ibukota Jakarta 13560

CHAPTER III
LITERATURE REVIEW

3.1. Web Programming

The World Wide Web (WWW) is a system of information identified by global identifiers (URLs) to access useful resources and consists of server sites worldwide that exchange data and information. Although often used interchangeably with the Internet, the two are different. The Internet is a global network of devices that enables activities like emailing and chatting, while the WWW is a network of servers over the Internet used to access websites. When you request a web page using a browser, your computer (client) communicates with a server to render the page in your browser. [2]

The process of creating web pages that can be accessed by internet visitors is called web programming. The standard used in this web programming is HTML. This standard is used so that people can read information under different circumstances. The essence of web programming is to create a web page according to standards so that people can receive the information on the page.

3.2. HTML

HTML stands for Hypertext Markup Language. HTML is a standard used in web programming so that people can understand the information on a web page. HTML is usually paired with javascript and css in order to make a more appealing design on a website. [1]

3.3. Enterprise Resource Planning

ERP stands for enterprise resource planning. ERP itself is software used to help companies in their daily work activities. Things like project management, parts ordering, delivery status, supply chain operations, can all be reviewed using this system. By integrating ERP systems into business processes, organizations can ensure smooth data flow and eliminate duplication. This system also functions as the central hub for end-to-end workflow. ERP provides transparency into production, logistics, and finance, and can be customized for various industries, making it very important for businesses of all sizes. [4]

3.4. Laravel

Laravel is a PHP-based framework useful for developing web applications. It is known for providing a solid structure in web development. With features like scheduled jobs, dependency injection, etc., Laravel can enhance developer productivity. This framework uses an MVC (Model-View-Controller) architecture. Its flexibility and powerful features make it an ideal choice for building web applications, such as those worked on in this practical work. [5]

3.5. Tailwind

Tailwind CSS is a framework with a utility-first approach. This approach allows developers to use predefined classes directly in their markup so they can create custom designs. This method simplifies the design process and supports consistency during UI development. Tailwind also supports responsive design and follows a mobile-first principle, which is one of the reasons why Tailwind is used in this practical work.[3]

[This page is purposefully left blank]

CHAPTER IV
SYSTEM ANALYSIS

4.1.   System Analysis

This chapter will cover the steps on applying Tailwind CSS to an already existing page in the ERP web app. This chapter itself will be divided into two. The first one being an overview of the application and the second one will explain on the process of identifying the UI of the specific pages that require updates. But before that, let's take a look of the overview of ERP itself.

The Enterprise Resource Planning (ERP) system is a comprehensive software solution implemented by PT. Uniteda Arkato to streamline and manage their daily business operations. As part of their initiative to integrate offices nationwide, this system consolidates various business processes such as project management, parts ordering, delivery status tracking, and supply chain operations into a single monitoring system. By centralizing these processes, the ERP system ensures a smooth data flow, eliminates duplication, and provides a single data source for the organization.

4.2.   Identifying Web Pages for Updates

The first step before implementing Tailwind CSS into the web pages is to determine which pages require a responsive design. Developers must prioritize specific pages instead of applying changes to all pages because the website is live and being actively used by employees. This means developers must focus on pages that are frequently accessed by employees using smartphones. Table 4.1

11

contains the list of pages that author have been assigned to work on.

Table 4.1 List of Pages with UI Problems

No	Page Name	User Interface Problem
1	Work Order (WO)	1. The layout of the filter is not responsive 2. Some of the elements is not properly aligned 3. The forms are not properly labelled when in mobile view
2	Create Work Order	1. Some of the elements are not properly aligned
3	Purchase Requisition (PR)	1. The layout of the filter is not responsive 2. Some of the elements is not properly aligned 3. The forms are not properly labelled when in mobile view
4	PR Operation	1. The layout of the filter is not responsive 2. Some of the elements is not properly aligned 3. The forms are not properly labelled when in mobile view
5	Catalogues	1. The layout of the filter is not responsive 2. Some of the elements is not properly aligned

No	Page Name	User Interface Problem
		3. The forms are not properly labelled when in mobile view
6	Asset	1. The layout of the filter is not responsive
		2. Some of the elements is not properly aligned
		3. The forms are not properly labelled when in mobile view
7	Daily Maintenance Report	1. The layout of the filter is not responsive
		2. Some of the elements is not properly aligned
		3. The forms are not properly labelled when in mobile view
8	Customer	1. The layout of the filter is not responsive
		2. Some of the elements is not properly aligned
		3. The forms are not properly labelled when in mobile view
9	TSD Report	1. The layout of the filter is not responsive
		2. Some of the elements is not properly aligned

No	Page Name	User Interface Problem
		3. The forms are not properly labelled when in mobile view
10	Location	1. The layout of the filter is not responsive 2. Some of the elements is not properly aligned 3. The forms are not properly labelled when in mobile view
11	Operation	1. The layout of the filter is not responsive 2. Some of the elements is not properly aligned 3. The forms are not properly labelled when in mobile view
12	Good Issue	1. The layout of the filter is not responsive 2. Some of the elements is not properly aligned 3. The forms are not properly labelled when in mobile view
13	Good Receipt	1. The layout of the filter is not responsive 2. Some of the elements is not properly aligned

No	Page Name	User Interface Problem
		3. The forms are not properly labelled when in mobile view
14	Report Incoming	1. The layout of the filter is not responsive 2. Some of the elements is not properly aligned 3. The forms are not properly labelled when in mobile view
15	Inventory Control (Non-Consignment)	1. The layout of the filter is not responsive 2. Some of the elements is not properly aligned 3. The forms are not properly labelled when in mobile view
16	Inventory Stock	1. The layout of the filter is not responsive 2. Some of the elements is not properly aligned 3. The forms are not properly labelled when in mobile view
17	Material Request	1. The layout of the filter is not responsive 2. Some of the elements is not properly aligned

No	Page Name	User Interface Problem
		3. The forms are not properly labelled when in mobile view
18	Outgoing Parts	1. The layout of the filter is not responsive 2. Some of the elements is not properly aligned 3. The forms are not properly labelled when in mobile view
19	Report Purchase Order	1. The layout of the filter is not responsive 2. Some of the elements is not properly aligned 3. The forms are not properly labelled when in mobile view
20	Report Purchase Order Outstanding	1. The layout of the filter is not responsive 2. Some of the elements is not properly aligned 3. The forms are not properly labelled when in mobile view
21	Report Purchase Requisition	1. The layout of the filter is not responsive 2. Some of the elements is not properly aligned

No	Page Name	User Interface Problem
		3. The forms are not properly labelled when in mobile view
22	Report Outgoing	1. The layout of the filter is not responsive 2. Some of the elements is not properly aligned 3. The forms are not properly labelled when in mobile view
23	Lead Time	1. The layout of the filter is not responsive 2. Some of the elements is not properly aligned 3. The forms are not properly labelled when in mobile view
24	Lead Time Approval PR	1. The layout of the filter is not responsive 2. Some of the elements is not properly aligned 3. The forms are not properly labelled when in mobile view
25	Lead Time PR to PO	1. The layout of the filter is not responsive 2. Some of the elements is not properly aligned

No	Page Name	User Interface Problem
		3. The forms are not properly labelled when in mobile view
26	Material Request Inventories	1. The layout of the filter is not responsive 2. Some of the elements is not properly aligned 3. The forms are not properly labelled when in mobile view
27	Report Operation	1. The layout of the filter is not responsive 2. Some of the elements is not properly aligned 3. The forms are not properly labelled when in mobile view
28	Quotations	1. The layout of the filter is not responsive 2. Some of the elements is not properly aligned 3. The forms are not properly labelled when in mobile view
29	Report TSD	1. The layout of the filter is not responsive 2. Some of the elements is not properly aligned

No	Page Name	User Interface Problem
		3. The forms are not properly labelled when in mobile view
30	Create Work Order Report	1. The layout of the form is not responsive 2. Some of the elements are not properly aligned

[This page is purposefully left blank]

CHAPTER V
SYSTEM IMPLEMENTATION

This chapter will go over the implementation of Tailwind CSS into the pages that have been mentioned in Chapter 4. It will detail the specific steps taken to integrate Tailwind CSS, including applying utility classes and testing the responsiveness of each page. Through these efforts, the goal is to enhance the user interface and provide a better experience for mobile users.

5.1. Implementation of Tailwind CSS

This part will cover the implementation of Tailwind CSS grouped on the problem found throughout the pages. It is not done per web page in order to shorten the length of this report and to not make it redundant. Each sub-chapter will include at least 2 pages each problem to better illustrate what the problem is and what the author did to fix the problem.

5.1.1. Unresponsive Filter Layout

Figure 5.1, 5.2, 5.3, and 5.4 are the examples of some of the pages that have unresponsive filter within the page itself.

Figure 5.1 Work Orders Page in Desktop View

Figure 5.2 Work Orders Page in Mobile View

Figure 5.3 Purchase Requisition Page in Desktop View

23
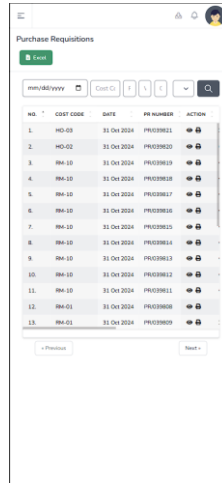
Figure 5.4 Purchase Requisition Page in Mobile View

It can be inferred from the Figures that the filter part above the table on each page becomes small when switched to a mobile view, making it hard to know what the user inputted inside the filter. Furthermore, the placeholder on each of the filter field also becomes unreadable. To fix this problem we can implement tailwind using these codes.

```
.
<div class="lg:flex items-center pb-6 ">
     @if(auth()->user()->hasRole('warehouse'))
         <form method="GET" action="{{ route('work-orders.index-warehouse') }}"
class="lg:flex items-end gap-2 m-0 grid grid-cols-2">
     @else
     <form method="GET" action="{{ route('work-orders.index') }}" class="lg:flex
items-end gap-2 m-0 grid grid-cols-2">
     @endif
.
.
.
```

Source Code 5.1 Snippet of Code for Work Orders Page

This source code 5.1 is to fix the layout of each element inside the filter div. It uses the classes like "lg:" to make sure the settings for the desktop views remains the same. Then, by using "grid" and "grid-cols" to achieve the responsive layout the user needs. Here are Figure 5.5 and 5.6 showing what the page looks like after using the code.

Figure 5.5 Work Orders Page on Mobile View After Applying Tailwind
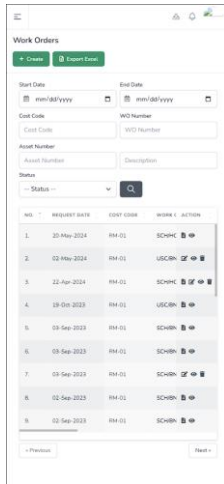
Figure 5.6 Purchase Requisition Page on Mobile View After Applying Tailwind

As seen in the Figure, now the layout of the filter is a lot tidier and comprehensible for the users.

5.1.2. Misaligned Elements
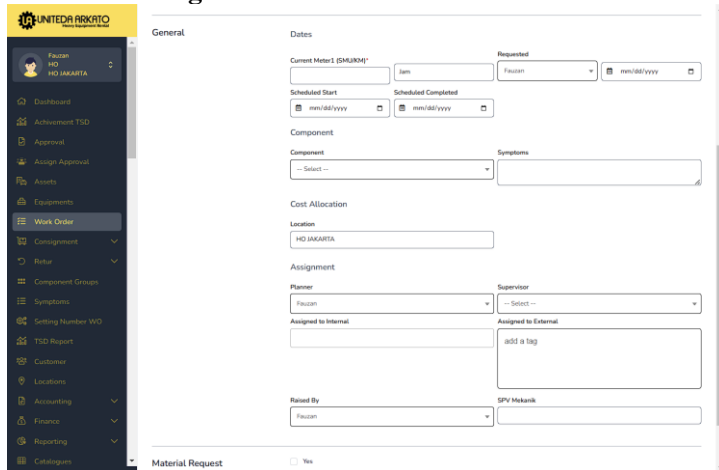
Figure 5.7 Create Work Orders Page on Desktop View
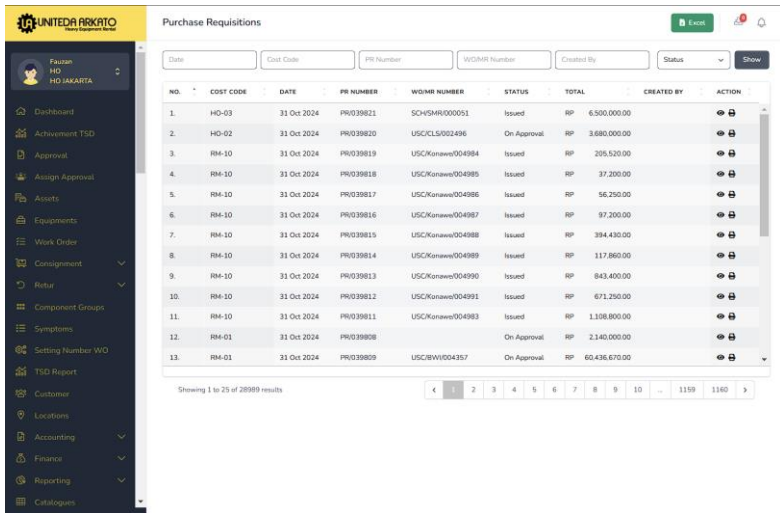
Figure 5.8 Purchase Requisitions Page on Desktop View

As seen in Figure 5.7 and 5.8, some elements are not properly aligned. For Create Work Order Page, in order to align those, the code can be updated as such.

```
.
.
.
<x-wire-input    required    label="{{__('Current    Meter1    (SMU/KM)')}}"
inputId="current_meter"
                        model="current_meter" value="{{ old('current_meter') }}"
class="current_meter0" />
            </div>
            <div class="flex-1 pb-2">
                        <x-wire-input-leading  inputId="current_meter_time"
model="current_meter_time"
                value="{{ old('current_meter_time') }}" class="timepicker">
                <x-slot name="leadingText">Jam</x-slot>
            </div>

        <div class="flex space-x-2 items-end">
          <div class="flex-1 pb-2">
```

```
                          <x-wire-select inputId="requested" label="{{__('Requested')}}"
model="requested"
                      class="input-select2">
                      <option value="">-- {{__('form.select')}} --</option>
.
.
```

Source Code 5.2 Create Work Order Page Code Before Code Update

```
.
.
<x-wire-input      required      label="{{__('Current      Meter1      (SMU/KM)')}}"
inputId="current_meter"
                          model="current_meter" value="{{ old('current_meter') }}"
class="current_meter0" />
              </div>
              <div class="flex-1">
                              <x-wire-input-leading  inputId="current_meter_time"
model="current_meter_time"
                      value="{{ old('current_meter_time') }}" class="timepicker">
                      <x-slot name="leadingText">Jam</x-slot>
                </x-wire-input-leading>
              </div>
           </div>

           <div class="flex space-x-2 items-end">
              <div class="flex-1">
                   <x-wire-select inputId="requested" label="{{__('Requested')}}"
model="requested"
                      class="input-select2">
                      <option value="">-- {{__('form.select')}} --</option>
.
.
.
```

Source Code 5.3 Create Work Order Page Code Before Code Update

In this code example the thing that I change is
deleting the "pb-2" class. This class is padding bottom and
the one responsible for making the element position isn't
properly aligned.
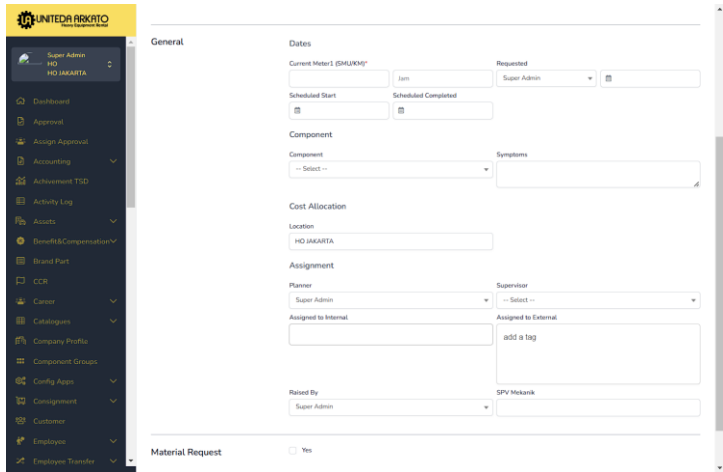
Figure 5.9 Create Work Order Page After Code Update

The Figure 5.9 shows the elements have properly aligned after erasing the bottom padding. For the Purchase Requisition page, the code can be updated as such.

```
</x-slot>
<x-container>
   <div class="pb-6 ">
      <form method="GET" action="{{ route('agreements.index') }}" class="flex items-
center gap-2 ">
         @csrf
         <div class="w-full sm:w-auto">
            <x-wire-input
               inputId="search"
               placeholder="{{__('Search')}}"
               value="{{ Request::get('search') }}"
               model="search"
               inputType="horizontal"
               class="w-full"
               style="width:300px"
            />
         </div>
         <div>
            <x-wire-input
               {{-- label="{{__('form.date')}}" --}}
               inputId="date"
```

```
        placeholder="yyyy-mm-dd"
        model="date"
        class="date"
        inputType="horizontal"
        value="{{ Request::get('date') }}"
        style="width:110px"
    />
    <x-wire-select
        inputId="accept"
        model="accept"
        inputType="horizontal"

        >
        <option value="">-- {{__('Select Accept')}} --</option>
         <option value="1" {{ Request::get('accept') == '1' ? 'selected="selected"' :
"}}>{{__('Yes')}}</option>
            <option value="0" {{ Request::get('accept') == '0' ? 'selected="selected"' :
"}}>{{__('No')}}</option>
        </x-wire-select>
    </div>
    <div>
        <x-button type="submit" hasColor="gray">
        <span class="hidden sm:block">{{__('button.btn_show')}}</span>
        <span class="block sm:hidden ">
```

Source Code 5.4 Purchase Requisition Page Code Before
Updating

```
                        </x-slot>
                        <x-container>
                        <div class="pb-6 ">
<form method="GET" action="{{ route('agreements.index') }}" class="sm:flex
                    items-end gap-2 grid grid-cols-2">
                              @csrf
                        <div class="w-full sm:w-auto">
                              <x-wire-input
                                inputId="search"
            label="<span class='sm:hidden'>{{   ('Search')}}</span>"
                        placeholder="{{   ('Search')}}"
                        value="{{ Request::get('search') }}"
                              model="search"
                                 />
                              </div>
                              <div>
                              <x-wire-input
            label="<span class='sm:hidden'>{{   ('form.date')}}</span>"
                              inputId="date"
                        placeholder="yyyy-mm-dd"
                              model="date"
```

30

```
                            class="date"
                    value="{{ Request::get('date') }}"
                         style="width:110px"
                                  />
                          <x-wire-select
                          inputId="accept"
                          model="accept"
            label="<span class='sm:hidden'>{{__('Select Accept')}}</span>"

                                  >
          <option value="">-- {{   ('Select Accept')}} --</option>
              <option value="1" {{ Request::get('accept') == '1'  ?
        'selected="selected"' : ''}}>{{   ('Yes')}}</option>
              <option value="0" {{ Request::get('accept') == '0'  ?
        'selected="selected"' : ''}}>{{   ('No')}}</option>
                          </x-wire-select>
                               </div>
                      <div class="mt-3 sm:mt-0">
              <x-button type="submit" hasColor="gray">
        <span class="hidden sm:block">{{   ('button.btn  show')}}</span>
                    <span class="block sm:hidden ">
```

Source Code 5.5 Purchase Requisition Page Code After Updating

In this code, the changes are applying the class "mt-3" which is margin top to properly align the elements and removing "inputType="horizontal"" as well as other hard style CSS such as "style="width:300px"". After applying these changes, the appearance of the Purchase Requisition page will look like Figure 5.10.

Figure 5.10 Purchase Requisition Page After Code Update

5.1.3. **Missing Label on Mobile View**

In some pages, when switched to the mobile view the label or placeholder on the filter form existing in said pages becomes unreadable. To fix this problem we can apply label by using tailwind and only show it when in mobile view. Figure 5.11, and figure 5.12 are examples of pages with this problem.
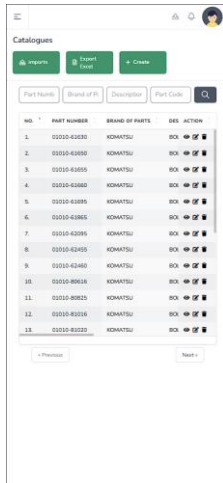
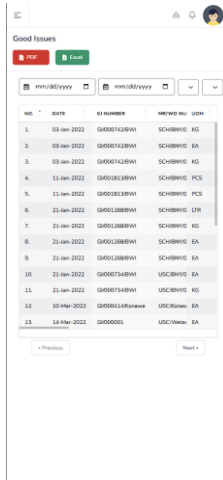*Figure 5.11 Catalogue Page on Mobile View*

Figure 5.12 Good Issue Page on Mobile View

As seen in Figure 5.11 and 5.12, the placeholder becomes unreadable in mobile view. This can cause

confusion for users because they wouldn't be able to differentiate the purpose of each filter form. To fix this we can apply the following.

```
<div>
    <x-wire-input-leading
      inputId="start_date"
      label="<span class='md:hidden'>{{__('Start Date')}}</span>"
      placeholder="{{__('Start Date')}}"
      model="start_date"
      value="{{ Request::get('start_date') }}"
      {{-- inputType="horizontal" --}}
    ><x-slot name="leadingText"><i class="far fa-calendar-alt"></i></x-slot></x-wire-input-leading>
</div>

<div>
    <x-wire-input-leading
      inputId="end_date"
      label="<span class='md:hidden'>{{__('End Date')}}</span>"
      placeholder="{{__('End Date')}}"
      model="end_date"
      value="{{ Request::get('end_date') }}"
      {{-- inputType="horizontal" --}}
    ><x-slot name="leadingText"><i class="far fa-calendar-alt"></i></x-slot></x-wire-input-leading>
</div>
        @if (Auth::user()->hasRole('superadmin')  ||  strtoupper(auth()->user()->locationType) == 'HO')
    <div>
    <x-wire-select
      inputId="warehouse_from"
      label="<span class='md:hidden'>{{__('Warehouse (From)')}}</span>"
      model="warehouse_from"
      class="cost_code0"
      >
    <div>
    <x-wire-select
      inputId="warehouse_to"
      label="<span class='md:hidden'>{{__('Warehouse (To)')}}</span>"
      model="warehouse_to"
      class="cost_code0"
      >
```

Source Code 5.6 Snippet of Good Issue Page Code

In this code, the variable label is used in each filter form to display labels above the form. Then the label is given class "md:hidden" to hide it when the screen width reaches 768 pixel. The class make it so the label would only appear in mobile view and not the desktop view. Thus, the updated appearance of these pages would be like Figure 5.13 and 5.14.

Figure 5.13 Catalogue Page in Mobile View After Update

Figure 5.14 Good Issue Page in Mobile View After Update

5.2. Implementation of Laravel

Aside of Tailwind CSS, while doing this project, author also do some of coding that involves Laravel. The purpose of this code is to sort the sidebar menu alphabetically. Figure 5.15 and 5.16 will show the difference between the old sidebar and the new sidebar.
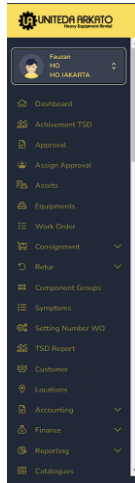
Figure 5.15 The Sidebar Before Sorting
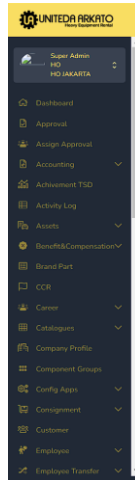
Figure 5.16 The Sidebar After Sorting

It can be seen in Figure 5.15 that the old sidebar is not alphabetically sorted. The sidebar pictured in Figure 5.16 is alphabetically sorted giving it a tidier layout.

[This page is purposefully left blank]

CHAPTER VI
EVALUATION

This chapter will cover the evaluation process after successfully implementing Tailwind CSS. The evaluation aims to ensure that the implemented design has fulfilled the efficiency that user needs based on the assessment of the supervisor.

6.1.   Purpose of Evaluation

This evaluation is commenced with the purpose of assessing the efficiency and effectiveness of the updated design on the ERP web app pages based on the supervisor's feedback.

6.2.   Evaluation Steps

Steps done when doing this evaluation are as the following:

1. Supervisor reviewing the updated web pages on various devices.
2. Assessing the design changes for visual and functional improvements.
3. Providing feedback on the efficiency and effectiveness of the new design.

6.3.   Evaluation Result

The results of the evaluation are based on the supervisor's feedback. Table 6.1 summarizes the evaluation results.

Table 6.1. Evaluation Result

Criteria	Result
The design has improved user experience and interface design	Achieved
Responsiveness of the pages on different devices	Achieved
Supervisor's assessment of the design changes	Achieved

However, despite the approval from the field supervisor, the changes made has not yet been implemented into the live website. This is due to the fact that the website is currently live and the field supervisor is concerned that if the developer team suddenly made a huge UI change the users will protest since they are not familiar with the newly designed User Interface.

CHAPTER VII
CONCLUSION AND SUGGESTION

7.1. Conclusion

In this practical work, Tailwind CSS was successfully implemented to improve the design and responsiveness of specific pages within the ERP web app. The main goal of this practical work, which is to enhance the responsiveness of the pages, particularly for employees who uses it in mobile view has certainly been achieved. This is based on the supervisor's agreement with the design that has been done by the author.

To sum up the process from the start, first of all it starts with identifying the pages that needed to have its UI updated due to responsiveness problem. Then by applying Tailwind CSS into these pages, making use of its utility classes, the responsiveness and UI elements of said pages can be improved. After that, the page is tested by the field supervisor to confirm that it has met the standard and behaviour of the users in PT. Uniteda Arkato.

The evaluation of this implementation was primarily based on the supervisor's feedback. The supervisor's assessment confirmed that the design changes were effective in enhancing the user experience and met the expected criteria for efficiency and functionality. This validation highlighted the importance of targeted updates and the positive impact of modern CSS frameworks like Tailwind on web application development. However, it has to be noted that the changes made has not yet been implemented into the live website.

7.2. Suggestion

Here are some suggestions of the things that the author think would be beneficial for this project:

1. User Feedback**:** Although the initial evaluation was supervisor-based, gathering direct feedback from end-users is crucial for further improvements.
2. Involve More Users: It has been said that the evaluation for this project is heavily relied on the field supervisor. Outside of that, even if there is user involved, it is still a lot less than needed for a proper user feedback, which will make a problem as the project goes.
3. Make a Mobile-First Approach When Building an Interface**:** Considering that the problem in this practical work arose from the fact that the developers not using a mobile-first approach, the author thinks that using this approach when building future features would surely be beneficial for the development time.

These conclusions and suggestions aim to guide future development efforts and maximize the benefits of the ERP system for PT. Uniteda Arkato.

BIBLIOGRAPHY

[1] J. Enterprise, HTML 5 MANUAL BOOK, Jakarta: PT Elex Media Komputindo, 2014.

[2] Budjianto, James Christian. 2020. Evolution of the World Wide Web. [ONLINE] available at: https://sis.binus.ac.id/2020/11/13/evolution-of-the-world-wide-web/ [Accessed on November 2024]

[3] Tailwind Labs Inc., 2024. Tailwind CSS Documentation. [ONLINE] available at: tailwindcss.com [Accessed on November 2024]

[4] Oracle. What is ERP? [ONLINE] available at: https://www.oracle.com/id/erp/what-is-erp [Accessed on November 2024]

[5] Laravel Holdings Inc. 2024. Laravel Documentation [ONLINE] available at: https://laravel.com/docs/11.x [Accessed on November 2024]

[This page is purposefully left blank]

AUTHOR'S BIOGRAPHY

Name   : Fauzan Ahmad Faisal
Date and Place of Birth : Jakarta, 28th June 2003
Gender   : Male
Phone Number   : +6285770341919
Email   : fauzan.afaisal@gmail.com

AKADEMIS
College   : Informatics Department –
                                             FTEIC, ITS
Batch   : 2021
Semester   : 7 (Seven)