

**WYDZIAŁ  
ELEKTROTECHNIKI  
I INFORMATYKI  
POLITECHNIKI RZESZOWSKIEJ**

**Jakub Turczyński**

**Usługi sieciowe w biznesie - Projekt**

**Aplikacja webowa do przeglądania zadłużeń kontrahentów**

Rzeszów, 2023

## 1. Cel projektu

Projekt "Aplikacja webowa do przeglądania zadłużeń kontrahentów" realizowany z użyciem technologii PHP i JavaScript ma na celu stworzenie efektywnego i użytkowniczego narzędzia do zarządzania i przeglądania informacji o zadłużeniach kontrahentów.

## 2. Funkcjonalności aplikacji

Poniżej wyszczególnione zostały główne funkcjonalności aplikacji:

- **Intuicyjny interfejs użytkownika:** Aplikacja powinna posiadać czytelny i łatwy w obsłudze interfejs użytkownika (UI), który umożliwi szybki dostęp do różnych funkcji aplikacji.
- **Integracja z bazą danych:** Aplikacja powinna być zintegrowana z bazą danych, która pozwoli na przechowywanie i zarządzanie informacjami o zadłużeniach kontrahentów.
- **Funkcjonalność wyszukiwania i filtrowania:** Aplikacja powinna oferować funkcje wyszukiwania i filtrowania, aby użytkownicy mogli łatwo znajdować specyficzne informacje o zadłużeniach.

### 3. Użyte technologie

Aplikacja korzysta z różnych technologii, aby zapewnić funkcjonalność i wydajność. Oto opis użytych technologii:

**MySQL:** Jest to system zarządzania relacyjnymi bazami danych (RDBMS), który używa języka SQL (Structured Query Language) do zarządzania danymi. W aplikacji "Webowerj" MySQL służy do przechowywania, odczytywania, modyfikowania i usuwania danych, takich jak informacje użytkowników, dane transakcji czy inne istotne dla aplikacji informacje. Jest to popularny wybór dla aplikacji webowych ze względu na jego niezawodność, skalowalność i wsparcie społeczności.

**PHP 8.1:** Jest to serwerowy język skryptowy używany głównie do tworzenia dynamicznych stron internetowych i aplikacji webowych. PHP 8.1 to najnowsza wersja, która oferuje wiele nowych funkcji i usprawnień wydajności w porównaniu do swoich poprzedników. W aplikacji "Webowerj", PHP jest wykorzystywane do przetwarzania logiki po stronie serwera, obsługi zapytań do bazy danych i generowania dynamicznych treści HTML na podstawie danych użytkownika i bazy danych.

**JavaScript (JS):** Jest to język programowania, który pozwala na tworzenie dynamicznych i interaktywnych stron internetowych. JavaScript jest wykorzystywany do manipulowania treścią na stronie, obsługi zdarzeń, walidacji danych i wielu innych zadań. Jest to kluczowy element interaktywności użytkownika w aplikacji webowej.

**jQuery:** Jest to szybka, mała i bogata w funkcje biblioteka JavaScript. Uproszcza ona takie zadania jak manipulacja dokumentem HTML, obsługa zdarzeń, animacja i Ajax, dzięki łatwej w użyciu API, która działa na wielu przeglądarkach. jQuery jest często wykorzystywana do uproszczenia złożonego kodu JavaScript i ułatwienia cross-browser compatibility.

**Bootstrap:** Jest to framework front-endowy do tworzenia responsywnych i mobilnie pierwszych stron internetowych. Bootstrap oferuje zbiór narzędzi do tworzenia interfejsów użytkownika z predefiniowanymi komponentami HTML i CSS, takimi jak formularze, przyciski, nawigacje i inne elementy interfejsu. Jest on używany do szybkiego tworzenia estetycznych i funkcjonalnych układów strony.

[illegible]

b) **client\_information**- przechowuje informacje o klientach na których dokumenty zostały wystawione.

[illegible]

Po utworzeniu tabeli wypełnione zostały przykładowymi danymi:

**Navigation**

Filter objects

- api
- debtcollect
  - Tables
    - client\_information
    - invoice\_summary
  - Views
  - Stored Procedures
  - Functions
- sys
- target
- tywent
- webserver
- webserver\_2022\_2023

Query 1   client\_information   invoice\_summary   invoice\_summary   invoice\_summary - Table   invoice\_summary   client\_information   **invoice\_summary**

1 • **SELECT \* FROM debtcollect.invoice\_summary;**

**Result Grid** | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: ☒

	invoice_id	document_number	issue_date	payment_due_date	payment_date	payment_method	amount_due	amount_paid	balance	remarks	company_id
1	FV000001		2024-01-15	2024-02-14	NULL	Przelew bankowy	1000.00	0.00	-1000.00		8
2	FV000002		2024-01-16	2024-02-15	2024-01-20	Przelew bankowy	2000.00	2000.00	0.00		2
3	FV000003		2023-12-17	2024-01-16	NULL	Gotówka	3000.00	0.00	-3000.00	Opóźnienie 2 dni	9
4	FV000004		2023-12-18	2024-01-17	2024-01-18	Karta kredytowa	4000.00	4000.00	0.00		6
5	FV000005		2024-01-19	2024-02-18	NULL	Przelew bankowy	1500.00	0.00	-1500.00	Opóźnienie 3 dni	3
6	FV000006		2023-11-20	2023-12-20	2023-12-21	Przelew bankowy	2500.00	2500.00	0.00		6
7	FV000007		2024-01-21	2024-02-20	NULL	Gotówka	3500.00	0.00	-3500.00	Bez uwag	3
8	FV000008		2023-12-22	2024-01-21	2024-01-25	Przelew bankowy	2200.00	2200.00	0.00		4
9	FV000009		2023-12-23	2024-01-22	NULL	Gotówka	1100.00	0.00	-1100.00	Opóźnienie 1 dni	3
10	FV000010		2024-01-24	2024-02-23	NULL	Przelew bankowy	500.00	0.00	-500.00	Brak opóźnień	3
11	FV000011		2023-11-25	2023-12-25	2023-12-30	Gotówka	600.00	600.00	0.00		3
12	FV000012		2024-01-26	2024-02-25	NULL	Przelew bankowy	700.00	0.00	-700.00	Opóźnienie 2 dni	6
13	FV000013		2023-11-27	2023-12-27	2024-01-03	Karta kredytowa	800.00	800.00	0.00		10
14	FV000014		2024-01-28	2024-02-27	NULL	Przelew bankowy	900.00	0.00	-900.00	Opóźnienie 1 dni	1
15	FV000015		2023-10-29	2023-11-29	2023-11-30	Gotówka	1000.00	1000.00	0.00		7
16	FV000016		2024-01-30	2024-02-28	NULL	Przelew bankowy	1100.00	0.00	-1100.00	Brak opóźnień	10
17	FV000017		2023-10-31	2023-11-30	2023-12-01	Karta kredytowa	1200.00	1200.00	0.00		8
18	FV000018		2024-01-01	2024-02-01	NULL	Przelew bankowy	1300.00	0.00	-1300.00	Opóźnienie 3 dni	9
19	FV000019		2023-12-02	2024-01-02	2024-01-04	Gotówka	1400.00	1400.00	0.00		2
20	FV000020		2024-01-03	2024-02-02	NULL	Przelew bankow	1500.00	0.00	-1500.00	Brak opóźnień	1

## 4.2 Kontroler przetwarzający żądania i przetwarzający dane

W celu umożliwienia naszej aplikacji skorzystania z danych znajdujących się w naszej bazie danych utworzony został kontroler ***DebtCollectionHandler.php***

W ciele tego kontrolera utworzona została funkcja wyciągająca wszystkie informacje o dokumentach znajdujących się w naszej bazie danych:

```
function dcGetDebtData()
{
    global $db;

    $debtData = $db->query('SELECT * FROM
debtcollect.invoice_summary where payment_date is null order by
payment_due_date desc;');

    return ['status' => "OK", 'data' => $debtData];
}
```

Teraz zwrócimy te dane w odpowiedzi na odpowiednie żądanie typu **POST**:

```
function main()
{
    global $dev;

    if (!headers_sent()) {
        header('Content-Type: application/json');
    }

    if ( !isset($_POST['dc']) ) {
        exit;
    }

    if ( isset($_POST['get_debt_data']) ) {
        $result = dcGetDebtData();
        echo json_encode( $result );
        exit;
    }
}
```

Dane z naszego kontrolera zwracane są w formacie **JSON**.

## 4.3 Wyświetlanie danych

Dane dokumentów wyświetlane są za pomocą tabeli. Do inicjalizacji tabeli posłużymy się pluginem do jQuery - **DataTables**

a) Tabela w pliku **index.php**:

```
<div class="container-fluid card box-shadow-1 my-3 px-1 px-md-2"
id="debtPanelTable">
  <div class="my-3">
    <table id="debtsTable" class="table">
      <thead>
        <tr>
          <th scope="col" class="text-center">ID Firmy</th>
          <th scope="col" class="text-center">ID
Dokumentu</th>
          <th scope="col" class="text-center">Dokument</th>
          <th scope="col" class="text-center">Data
wystawienia</th>
          <th scope="col" class="text-center">Wartość</th>
          <th scope="col" class="text-center">Zapłacił</th>
          <th scope="col" class="text-center">Saldo</th>
          <th scope="col" class="text-center">Termin</th>
          <th scope="col" class="text-center">Dni po
terminie</th>
          <th scope="col" class="text-center">Status</th>
          <th scope="col" class="text-center"><i class="bi
bi-three-dots"></i></th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>
            <div class="d-flex justify-content-center">
              <div class="spinner-border"
role="status">
                <span
class="visually-hidden">Loading...</span>
              </div>
            </div>
          </td>
          [ ... reszta komórek ... ]
        </tr>
      </tbody>
    </table>
  </div>
```

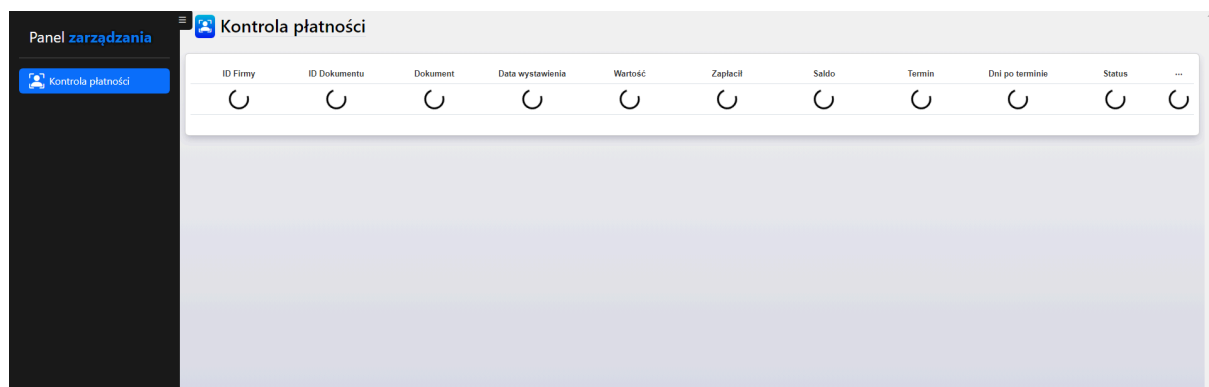
</div>

b) Inicjalizacja tabeli w pliku **debtcollection.js** przy pomocy pluginu **DataTables**:

```
$('#debtsTable').dataTable({
  createdRow: function (row, data) {
    if (data.class != null) {
      $(row).addClass(data.class);
    }
    if (data.style != null) {
      $(row).attr("style", data.style);
    }
  },
  ordering: false,
  autoWidth: false,
  columns: [
    {data: 'company_id', width: "7%"},
    {data: 'invoice_id', width: "8%"},
    {data: 'dokument', width: "13%"},
    {data: 'data_wys', width: "10%"},
    {data: 'wartosc', width: "9%"},
    {data: 'zaplacil', width: "9%"},
    {data: 'saldo', width: "9%"},
    {data: 'termin', width: "8%"},
    {data: 'opoznienie', width: "10%"},
    {data: 'ost_akcja', width: "12%"},
    {data: 'wiecej', width: "5%"}
  ],
  rowId: function(d) {
    return 'id_' + d.raw_inv_id;
  },
  columnDefs: [{
    targets: '_all',
    className: 'align-middle text-center p-1'
  }],
  language: {
    //Definicje tłumaczeń
  },
  pageLength: 50
});
```



Na ten moment nasz widok prezentuje się następująco:



Jak widać, tabela jest pusta. W celu pobrania danych wyślemy żądanie typu **POST** do naszego utworzonego wcześniej kontrolera ***DebtCollectionHandler.php***:

```
function getDebtsData(refresh = true) {
    $.post("handler/DebtCollectionHandler.php", {
        dc: true,
        get_debt_data: true
    }, function (response) {
        if (response != null && response != 'null') {
            console.log(response);
            try {
                if (response.status === "OK") {
                    debtsData.length = 0;
                    debtsData = response.data;
                }
            } catch (error) {
                console.log(error);
            }
        }

        if (refresh) {
            if (!$.fn.DataTable.isDataTable('#debtsTable')) {
                initializeDebtsTable();
            } else {
                setTimeout(updateDebtsTable, 100);
            }
            mainLoaderOff();
        }
    });
}
```

Funkcja ta wywołuje w swoim ciele również inne funkcje zajmujące się przetwarzaniem otrzymanych z serwera danych przed ich ostatecznym wyświetleniem w tabeli.

Wywoływana jest od razu po załadowaniu się naszej strony:

```
$(document).ready(function() {  
    //GET DATA  
    getDebtsData();  
})
```

W tym momencie nasza aplikacja wyświetla już interesujące nas dane o zadłużeniach kontrahentów i czasie opóźnienia płatności dla poszczególnych dokumentów:

Panel zarządzania

Kontrola płatności

Pokaż: 50

wyników na stronę

Wyszukaj:

ID Firmy	ID Dokumentu	Dokument	Data wystawienia	Wartość	Zapłacił	Saldo	Termin	Dni po terminie	Status	...
10	16	FV000016	2024-01-30	1100.00 zł	0.00 zł	-1100.00 zł	2024-02-28	-32 dni		▼
1	14	FV000014	2024-01-28	900.00 zł	0.00 zł	-900.00 zł	2024-02-27	-31 dni		▼
6	12	FV000012	2024-01-26	700.00 zł	0.00 zł	-700.00 zł	2024-02-25	-29 dni		▼
3	10	FV000010	2024-01-24	500.00 zł	0.00 zł	-500.00 zł	2024-02-23	-27 dni		▼
3	7	FV000007	2024-01-21	3500.00 zł	0.00 zł	-3500.00 zł	2024-02-20	-24 dni		▼
3	5	FV000005	2024-01-19	1500.00 zł	0.00 zł	-1500.00 zł	2024-02-18	-22 dni		▼
6	34	FV000034	2024-01-17	2900.00 zł	0.00 zł	-2900.00 zł	2024-02-16	-20 dni		▼
8	1	FV000001	2024-01-15	1000.00 zł	0.00 zł	-1000.00 zł	2024-02-14	-18 dni		▼
5	32	FV000032	2024-01-15	2700.00 zł	0.00 zł	-2700.00 zł	2024-02-14	-18 dni		▼
4	30	FV000030	2024-01-13	2500.00 zł	0.00 zł	-2500.00 zł	2024-02-12	-16 dni		▼
8	28	FV000028	2024-01-11	2300.00 zł	0.00 zł	-2300.00 zł	2024-02-10	-14 dni		▼
7	22	FV000022	2024-01-05	1700.00 zł	0.00 zł	-1700.00 zł	2024-02-04	-8 dni		▼
1	20	FV000020	2024-01-03	1500.00 zł	0.00 zł	-1500.00 zł	2024-02-02	-6 dni		▼
9	18	FV000018	2024-01-01	1300.00 zł	0.00 zł	-1300.00 zł	2024-02-01	-5 dni		▼
3	9	FV000009	2023-12-23	1100.00 zł	0.00 zł	-1100.00 zł	2024-01-22	5 dni		▼
9	3	FV000003	2023-12-17	3000.00 zł	0.00 zł	-3000.00 zł	2024-01-16	11 dni		▼
3	24	FV000024	2023-12-07	1900.00 zł	0.00 zł	-1900.00 zł	2024-01-06	21 dni		▼
5	26	FV000026	2023-11-09	2100.00 zł	0.00 zł	-2100.00 zł	2023-12-09	49 dni		▼

Pokazuje 1-18 na 18 wyników.

Poprzednia

1

Następna

## 4.4 Rozszerzenie funkcjonalności aplikacji

Rozszerzymy działanie naszej aplikacji aby wyświetlała ona szczegółowe informacje o kontrahencie na którego wystawiony jest dokument, a także o samym dokumencie.

Rozpoczniemy od stworzenia funkcji pobierającej informację o kontrahencie zarówno w naszym kontrolerze, jak i kodzie po stronie klienta:

### a) *DebtCollectionHandler.php*

*Utworzenie funkcji pobierającej dane o kontrahencie z bazy danych:*

```
function dcGetContractorData($companyId)
{
    global $db;

    $contractorData = $db->query("SELECT * FROM
debtcollect.client_information where company_id =
'{$companyId}';", true);

    return ['status' => "OK", 'data' => $contractorData];
}
```

*Modyfikacja funkcji main():*

```
function main()
{
    global $dev;

    if (!headers_sent()) {
        header('Content-Type: application/json');
    }

    if ( !isset($_POST['dc']) ) {
        exit;
    }

    if ( isset($_POST['get_debt_data']) ) {
        $result = dcGetDebtData();
        echo json_encode( $result );
        exit;
    } else if ( isset($_POST['get_contractor_data']) &&
isset($_POST['company_id']) && is_numeric($_POST['company_id'])) {
        $result = dcGetContractorData($_POST['company_id']);
        echo json_encode( $result );
        exit;
    }
}
```

b) *debtcollection.js*

```
function getContractorData(companyId) {
  return new Promise((resolve, reject) => {
    $.post("handler/DebtCollectionHandler.php", {
      dc: true,
      get_contractor_data: true,
      company_id: companyId
    }, function (response) {
      if (response != null && response !== 'null') {
        try {
          if (response.status === "OK") {
            resolve( response.data );
          } else {
            reject(response);
          }
        } catch (error) {
          reject(error);
        }
      }
    });
  });
}
```

Teraz utworzymy funkcję która zainicjalizuje akcję w odpowiedzi na zdarzenie kliknięcia w ikonę “strzałeczki” znajdującej się w ostatniej komórce w naszej tabeli. Jako odpowiedź na zdarzenie pobierzemy informację o kontrahencie przypisanym do danego dokumentu i wyświetlimy przetworzone dane zwracane z funkcji ***additionalAuctionDataFormat()***

### ***debtcollection.js***

```
//Events
function eventOnRowExpand() {
    $('#debtsTable').on('click', 'div.dc-show-more', function () {
        var table = $('#debtsTable');
        var dataTable = table.DataTable();
        var tr = $(this).closest('tr');
        var row = dataTable.row( tr );

        //Hide previous row
        var hideTr = table.find('.dt-hasChild');
        var hideRow = dataTable.row( hideTr );

        $('.chevron-selected').removeClass('chevron-selected');

        if ( hideRow.child.isShown() ) {

hideTr.find(".bi-chevron-up").removeClass("bi-chevron-up").addClasses("bi-chevron-down");
            $('div.slider', hideRow.child()).slideUp( 'fast',
function() {
                hideRow.child.hide();
                hideTr.removeClass('shown');
            } );

            //Expand/shrink
            if ( row.child.isShown() ) {
                $(this).html('<i class="bi bi-chevron-down chevron-selected"></i>');
                $('div.slider', row.child()).slideUp( 'fast',
function() {
                    row.child.hide();
                    tr.removeClass('shown');
                } );
            }
            else
            {
                let debtData =
getDebtDataFromId(row.data().raw_inv_id);
```



```
 <td><span class="fw-500">Adres:</span></td>  <td>`+cData.address+`</td>  <td><span class="fw-500">NIP:</span></td>  <td>`+cData.tax_id+`</td> </tr> <tr>  <td><span class="fw-500">Sposób płatności:</span></td>  <td>`+cData.payment_method+`</td>  <td><span class="fw-500">Limit kredytu:</span></td>  <td>`+cData.credit_limit+` <small>zł</small></td> </tr> <tr>  <td><span class="fw-500">Termin:</span></td>  <td>`+cData.payment_term_days+` <small>dni</small></td>  <td><span class="fw-500">Tolerancja:</span></td>  <td>`+cData.payment_tolerance_days+` <small>dni</small></td> </tr> <tr>  <td><span class="fw-500">Email:</span></td>  <td>`+cData.email+`</td>  <td><span class="fw-500">Telefon:</span></td>  <td>`+cData.phone_number+`</td> </tr> <tr>  <td><span class="fw-500">Obszar:</span></td>  <td>`+cData.area+`</td>  <td><span class="fw-500">Transport:</span></td>  <td>`+cData.transport+`</td> </tr> <tr>  <td><span class="fw-500">Fax:</span></td>  <td>`+cData.fax+`</td> </tr> </tbody> | | | | | | | | | | | | | | | | | | | | | |
```

```

        </table>
    </div>
</div>
</div>`;

// Invoice html
let invoiceHtml = `
    <h5>Szczegóły dokumentu</h5>
    <div class="border border-2 bg-white rounded-1 p-2">
        <div class="row align-items-start
dc-inv-info-container">
            <div class="col">
                <h1 class="lead
text-black">`+dData.document_number+' <strong title="ID
dokumentu">('+dData.invoice_id+')</strong>'+`</h1>
                <table class="table additional-table table-sm
mb-0 fs-14p">
                    <tbody>
                        <tr>
                            <td><span class="fw-500">ID
dokumentu:</span></td>
                            <td>`+dData.invoice_id+`</td>
                            <td><span class="fw-500">Numer
referencyjny:</span></td>
                            <td>`+dData.document_number+`</td>
                        </tr>
                        <tr>
                            <td><span class="fw-500">Data
wystawienia:</span></td>
                            <td>`+dData.issue_date+`</td>
                            <td><span
class="fw-500">Wartość:</span></td>
                            <td>`+dData.amount_due+`</td>
                        </tr>
                        <tr>
                            <td><span class="fw-500">Sposób
płatności:</span></td>
                            <td>`+dData.payment_method+`</td>
                            <td><span class="fw-500">Termin
płatności:</span></td>
                            <td class="fw-500 ${opoznienie > 0
? 'text-danger' : 'text-success'}">`+dData.payment_due_date+`</td>
                        </tr>
                        <tr>
                            <td><span class="fw-500">Opłacona
wartość:</span></td>
                            <td>`+dData.amount_paid+`</td>

```



```

                <td><span
class="fw-500">Saldo:</span></td>
                <td class="fw-500 ${0 >
dData.balance ? 'text-danger' : ''}">`+dData.balance+`</td>
            </tr>
            <tr>
                <td><span
class="fw-500">Uwagi:</span></td>
                <td>`+dData.remarks+`</td>
            </tr>
        </tbody>
    </table>
</div>
</div>
</div>`;

return '<div class="slider bg-light">' +
    '<div class="container-fluid py-2"
id="kon-'+cData.company_id+'-expand">' +
        '<div class="row align-items-start">' +
            '<div class="col-md-6">' +
                contractorHtml +
            '</div>' +
            '<div class="col-md-6">' +
                invoiceHtml +
            '</div>' +
        '</div>' +
    '</div>' +
    '</div>';
}

```

Po tych modyfikacjach możemy przeglądać szczegółowe informacje o wybranych dokumentach:

Panel zarządzania

Kontrola płatności

Pokaż: 50

wyników na stronę

Wyszukaj:

ID Firmy	ID Dokumentu	Dokument	Data wystawienia	Wartość	Zapłacił	Saldo	Termin	Dni po terminie	Status	...
19	16	FV000016	2024-01-30	1100.00 zł	0.00 zł	-1100.00 zł	2024-02-28	-32 dni		▼
1	14	FV000014	2024-01-28	900.00 zł	0.00 zł	-900.00 zł	2024-02-27	-31 dni		▼
6	12	FV000012	2024-01-26	700.00 zł	0.00 zł	-700.00 zł	2024-02-25	-29 dni		▲

Dane kontrahenta

Firma Budowlana Buildit (6)

Adres:

ul. Mursarska 5, 02-295 Warszawa

NIP:

9876543210

Sposób płatności:

Przelew bankowy

Limit kredytu:

60000.00 zł

Termin:

30 dni

Tolerancja:

5 dni

Email:

zamowienia@buildit.pl

Telefon:

22-987-65-43

Obszar:

WB-11

Transport:

BUILD-LOG

Fax:

22-987-65-44

Szczegóły dokumentu

FV000012 (12)

ID dokumentu:

12

Numer referencyjny:

FV000012

Data wystawienia:

2024-01-26

Wartość:

700.00

Sposób płatności:

Przelew bankowy

Termin płatności:

2024-02-25

Opłacona wartość:

0.00

Saldo:

-700.00

Uwagi:

Opóźnienie 2 dni

3

9

FV000009

2023-12-23

1100.00 zł

0.00 zł

-1100.00 zł

2024-01-22

5 dni

▲

Dane kontrahenta

GastroMix Sp.j. (3)

Adres:

pl. Smakoszy 3, 31-416 Kraków

NIP:

2345678901

Sposób płatności:

Karta kredytowa

Limit kredytu:

25000.00 zł

Termin:

45 dni

Tolerancja:

7 dni

Email:

info@gastromix.pl

Telefon:

12-345-67-89

Obszar:

KR-03

Transport:

GASTRO-DEL

Fax:

Szczegóły dokumentu

FV000009 (9)

ID dokumentu:

9

Numer referencyjny:

FV000009

Data wystawienia:

2023-12-23

Wartość:

1100.00

Sposób płatności:

Gotówka

Termin płatności:

2024-01-22

Opłacona wartość:

0.00

Saldo:

-1100.00

Uwagi:

Opóźnienie 1 dni

9

3

FV000003

2023-12-17

3000.00 zł

0.00 zł

-3000.00 zł

2024-01-16

11 dni

▼

## 5. Wnioski i spostrzeżenia

Podczas pracy nad projektem "Aplikacja webowa do przeglądania zadłużeń kontrahentów" zgłębiłem wiele aspektów tworzenia aplikacji webowych, co było dla mnie cennym źródłem praktycznej wiedzy. Ten projekt nie tylko pozwolił mi zastosować teoretyczne umiejętności, ale również nauczył mnie, jak ważne jest efektywne wykorzystanie różnych technologii.

Jednym z kluczowych elementów, który szczególnie wpłynął na mój rozwój, było zrozumienie znaczenia intuicyjnego interfejsu użytkownika. Wykorzystanie Bootstrap do stworzenia UI uświadomiło mi, jak ważne jest, aby interfejs był nie tylko estetycznie przyjemny, ale przede wszystkim prosty i intuicyjny w obsłudze. Ta lekcja podkreśliła dla mnie wartość projektowania z myślą o końcowym użytkowniku, co jest niezbędne w efektywnych aplikacjach biznesowych.

Wybór MySQL jako systemu zarządzania bazą danych był kolejnym ważnym aspektem projektu. Praca z tym systemem pozwoliła mi zrozumieć, jak istotne jest wykorzystanie stabilnych i sprawdzonych technologii w zarządzaniu dużymi ilościami danych. Doświadczenie to nauczyło mnie, że odpowiedni wybór technologii bazodanowych jest kluczowy dla zapewnienia niezawodności i wydajności aplikacji.

Zastosowanie PHP 8.1 dało mi możliwość przekonania się o zaletach wykorzystywania nowoczesnych funkcji tego języka. Używanie PHP do obsługi logiki serwerowej i tworzenia dynamicznych treści HTML okazało się być bardzo efektywne, co uświadomiło mi znaczenie aktualizowania wiedzy i umiejętności w zakresie wykorzystywanych technologii.

Praca z JavaScript i jQuery była również istotnym elementem projektu. Używanie tych technologii do tworzenia dynamicznych i interaktywnych elementów interfejsu ujawniło mi, jak mogą one znacząco poprawić doświadczenie użytkownika. Ta część projektu pokazała mi również wartość elastyczności w wyborze narzędzi do realizacji określonych funkcjonalności w aplikacji.

Podsumowując, realizacja tego projektu była niezwykle pouczająca. Dała mi ona dogłębne zrozumienie złożoności tworzenia aplikacji webowych i znaczenia prawidłowego doboru technologii oraz projektowania zorientowanego na użytkownika. Ta praktyczna wiedza z pewnością okaże się nieoceniona w mojej przyszłej karierze jako programista.