

Code Dokumentation Dennis Gordeev (Lead Management)

(Matrikelnummer: 4328420)

leads.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>CRM-Dashboard</title>
  <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">
  <link rel="stylesheet" href="styles/style.css">
  <link rel="stylesheet" href="styles/lead.css">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/font-awesome/6.2.0/css/all.min.css">
  <script src="scripts/leads.js"></script>
  <script src="scripts/getleads.js"></script>
  <script src="scripts/script.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
</head>
```

head:

- Einbindung von Bootstrap, dem übergreifenden Stylesheet (style.css) und einer Font für das allgemeine Design der Seite.
- Einbindung eines seitenspezifischen Stylesheets (lead.css).
- Einbindung von JavaScript-Skripten (leads.js, getleads.js, script.js (allgemeines Skript) und Bibliothek chart.js).

```
<main>

  <button id="openLeadFormBtn" class="new-lead-button">Neuen Lead hinzufügen</button>

  <div id="leadModal" class="modal">
    <div id="modal-content">
      <span class="close">&times</span>
      <form id="leadForm" action="index.php" method="post">
        <label for="first_name">Vorname:</label>
        <input type="text" id="first_name" name="firstName" required>
        <label for="last_name">Nachname:</label>
        <input type="text" id="last_name" name="lastName" required>
        <label for="email">Email:</label>
        <input type="text" id="email" name="email" required>
        <label for="phone">Telefonnummer:</label>
        <input type="number" id="phone" name="phone">
        <label for="company">Firma:</label>
        <input type="text" id="company" name="company" required>
        <button type="submit">Lead speichern</button>
      </form>
    </div>
  </div>
```

main:

- Darstellen des „New Lead Button“ für das Erzeugen eines neuen Eintrages.
- Erstellen eines Formulars mittels eines Modal-Screens und eines Form für das Eintragen der Daten.
- Weitergabe der Daten an die Datenbank durch „POST“ Übergabe an „index.php“.

```

<div class="leads">
  <h2>Leads</h2>
  <div class="search-wrapper">
    <!-- <i class="fas fa-search search-icon"></i> -->
    <input type="text" id="leadsSearchInput" class="search-input" placeholder="Nach Lead suchen">
    <button id="searchButton">Suchen</button>
  </div>
  <table id="leadsList">
    </table>
</div>
</main>

```

- Erzeugen der Klasse „leads“ zur Anzeige der Tabelle „leadsList“ mit den Lead-Daten aus der Datenbank.
- Erzeugen der Suchfunktion mit der Klasse „search-wrapper“

index.php:

```

if(isset($_SERVER["REQUEST_METHOD"]) && $_SERVER["REQUEST_METHOD"] === "POST") {
    $servername = "sql303.infinityfree.com";
    $username = "if0_36780543";
    $password = "if0_36780543";
    $dbname = "if0_36780543_crm_database";

    require_once('db_connect.php');
    $conn = new mysqli($servername, $username, $password, $dbname);

    if ($conn->connect_error) {
        die("Verbindung zur Datenbank fehlgeschlagen: " . $conn->connect_error);
    }
}

```

- Testen, ob eine „POST“-Anfrage existiert.
- Herstellen der Verbindung zur Datenbank, indem die Anmeldedaten an eine neue MySQL Verbindung weitergegeben werden.
- Wenn die Verbindung fehlschlägt, wird eine Error Message in die Konsole geschrieben.

```

$first_name = $_POST["firstName"];
$last_name = $_POST["lastName"];
$email = $_POST["email"];
$phone = $_POST["phone"];
$company = $_POST["company"];

$sql = "INSERT INTO if0_36780543_crm_database.leads (first_name, last_name, email, phone, company) VALUES (?, ?, ?, ?, ?)";
$stmt = $conn->prepare($sql);
$stmt->bind_param("sssss", $first_name, $last_name, $email, $phone, $company);
$stmt->execute();

if ($stmt->error) {
    echo "Fehler beim Einfügen des Leads: " . $stmt->error;
} else {
    echo "Lead erfolgreich gespeichert!";
}
$stmt->close();

```

- Übertragung der Daten aus dem Formular der leads.html an die Felder der Datenbank.
- Definieren des SQL-Insert-Statements, um die Daten in die leads Tabelle der Datenbank einzufügen.
- „Binden“ der Form-Parameter an die richtigen Bezeichnungen der Felder der Datenbanktabelle.
- Ausführen des SQL-Statements (\$stmt->execute());
- Überprüfung, ob das Einfügen der Daten erfolgreich war oder nicht und anschließendes Schließen der Connection.

getleads.js:

```

document.addEventListener('DOMContentLoaded', function () {
    fetch('get_leads.php')
        .then(response => response.json())
        .then(data => {
            const leadsTable = document.getElementById('leadsList');
            const leadsSearchInput = document.getElementById('leadsSearchInput');
            const tableHeaderRow = document.createElement('tr');
            const headerList = ['First Name', 'Last Name', 'Email', 'Company'];
            headerList.forEach(header => {
                const tableHeader = document.createElement('th');
                tableHeader.textContent = header;
                tableHeaderRow.appendChild(tableHeader);
            });
        });
});

```

- Erstellen einer Funktion zur Übertragung der Daten aus der Datenbanktabelle an die Tabelle auf der leads.html.
- Daten der get_leads.php werden abgerufen („fetch“).
- Erstellen der einzelnen Tabellen Elemente (leadsTable, leadsSearchInput, tableHeader, ...).

```

leadsTable.appendChild(tableHeaderRow);
data.forEach(lead => {
  const tableRow = document.createElement('tr');
  const tableDataFirstName = document.createElement('td');
  tableDataFirstName.textContent = lead.first_name;
  tableRow.appendChild(tableDataFirstName);
  const tableDataLastName = document.createElement('td');
  tableDataLastName.textContent = lead.last_name;
  tableRow.appendChild(tableDataLastName);
  const tableDataEmail = document.createElement('td');
  tableDataEmail.textContent = lead.email;
  tableRow.appendChild(tableDataEmail);
  const tableDataCompany = document.createElement('td');
  tableDataCompany.textContent = lead.company;
  tableRow.appendChild(tableDataCompany);
  leadsTable.appendChild(tableRow);
});

```

- Binden der Elemente aus der Datenbanktabelle an die erstellten Tabellen Elemente.

```

leadsSearchInput.addEventListener('input', function () {
  const searchTerm = leadsSearchInput.value.toLowerCase();
  const tableRows = leadsTable.querySelectorAll('tr:not(:first-child)');

  tableRows.forEach(row => {
    const tableData = row.querySelectorAll('td');
    let rowVisible = false;
    tableData.forEach(cell => {
      const cellText = cell.textContent;
      if (cellText.toLowerCase().includes(searchTerm)) {
        rowVisible = true;
        cell.innerHTML = cellText.replace(searchTerm, `<mark>${searchTerm}</mark>`);
      } else {
        cell.innerHTML = cellText;
      }
    });
    row.style.display = rowVisible ? '' : 'none';
  });
});

```

- Einbinden der Suchfunktion in die erstellte Tabelle.
- Erstellen der Markierfunktion, die den gesuchten Begriff einfärbt.

get_leads.php:

```
<?php

// error_reporting(E_ALL);
// ini_set('display_errors', 1);

$servername = "sql303.infinityfree.com";
$username = "if0_36780543";
$password = "QWERTYuiop1234567890";
$dbname = "if0_36780543_crm_database";

require_once('db_connect.php');
$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Verbindung zur Datenbank fehlgeschlagen: " . $conn->connect_error);
}

header('Content-Type: application/json');

$sql_select = "SELECT * FROM if0_36780543_crm_database.leads";
$result = $conn->query($sql_select);

if($result->num_rows > 0) {
    $leads = array();
    while($row = $result->fetch_assoc()) {
        $leads[] = $row;
    }
    echo json_encode($leads);
} else {
    echo json_encode(array());
}

$conn->close();

?>
```

- Erneutes Herstellen der Verbindung zur Datenbank.
- Definieren eines SQL-SELECT-Statements, welches alle Daten aus der leads-Datenbanktabellen abfragt (-> SELECT * FROM ...).
- Durchführen des SQL-Statements und Abfangen der Daten in ein Array.
- Schließen der Connection.

lead.css:

```
.modal {
  display: none;
  position: fixed;
  z-index: 100;
  left: 0;
  top: 0;
  width: 100%;
  height: 100%;
  background-color: rgba(0,0,0,0.5);
}

#modal-content {
  background-color: #fefefe;
  margin: 15% auto;
  padding: 20px;
  border: 1px solid #ccc;
  width: 60%;
}

.close {
  color: #aaa;
  float: right;
  font-size: 28px;
  font-weight: bold;
}

.close:hover,
.close:focus {
  color: black;
  text-decoration: none;
  cursor: pointer;
}
```

```
.new-lead-button {
  background-color: #007bff;
  color: #fff;
  border: none;
  padding: 20px 40px;
  font-size: 18px;
  border-radius: 5px;
  cursor: pointer;
  transition: background-color 0.3s;
  display: block;
  margin: 20px auto;
}

.new-lead-button:hover {
  background-color: #0056b3;
}
```

- Design und Positionierung des Formulars auf der leads.html (.modal und #modal-content).
- Design und Positionierung des Close-Buttons und Farbänderung beim „Hovern“ oder Auswählen des Buttons.
- Design und Positionierung des Buttons zur Erstellung eines neuen Leads und Farbänderung beim „Hovern“.

```

#leadForm label {
  display: block;
  margin-bottom: 10px;
}

#leadForm input[type="text"] {
  width: calc(100% - 20px);
  padding: 10px;
  margin-bottom: 20px;
  border: 1px solid #ccc;
  border-radius: 5px;
}

#leadForm button[type="submit"] {
  background-color: #007bff;
  color: #fff;
  padding: 10px 20px;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}

#leadForm button[type="submit"]:hover {
  background-color: #0056b3;
}

.lead {
  height: auto;
  width: 875px;
  background-color: #f0f0f0;
  border: 1px solid #ccc;
  border-radius: 5px;
  padding: 20px;
  margin: 20px auto 10px;
  position: relative;
}

```

```

.lead h2 {
  margin-top: 0;
}

#leadsList {
  list-style-type: none;
  padding: 0;
}

#leadsList li {
  margin-bottom: 10px;
}

.lead table {
  width: 100%;
  border-collapse: collapse;
}

.lead table th,
.lead table td {
  padding: 10px;
  border: 1px solid #ddd;
  text-align: left;
}

.lead table th {
  background-color: #f1f1f1;
  font-weight: bold;
}

```

- Positionierung des Labels des Formulars.
- Positionierung und Darstellung der Eingabe für Text im Formular.
- Positionierung und Design des „Submit“-Buttons und Farbänderung beim „Hovern“.
- Positionierung und Design der „lead“-Klasse.
- Positionierung der Überschrift der Tabelle.
- Positionierung und Design der Tabelle und der einzelnen Spalten und Reihen.

- Positionierung und Design der Suchleiste.
- Änderung der Farbe bei Auswahl der Suchleiste.
- Einstellen der Farbe, mit welcher der gesuchte Text markiert werden soll.

```
.search-wrapper {
  position: relative;
}

.search-input {
  background-color: #f2f2f2;
  border: 1px solid #ccc;
  padding: 5px 10px;
  border-radius: 5px;
  width: 100%;
  box-sizing: border-box;
  transition: all 0.2s ease-in-out;
}

.search-input:focus {
  background-color: #e0e0e0;
}

.mark {
  background-color: yellowgreen;
  color: black;
}
```

Ergebnis:

The image shows a web interface for managing leads. At the top, there is a blue button labeled 'Neuen Lead hinzufügen'. Below it is a section titled 'Leads' with a search bar containing the text 'Nach Lead suchen'. A modal window is open, showing a form with the following fields: Vorname, Nachname, Email, Telefonnummer, and Firma. Each field has a corresponding input box. At the bottom of the modal is a blue button labeled 'Lead speichern'.

Neuen Lead hinzufügen

Leads			
<input type="text" value="gordeev"/> <input type="button" value="Suchen"/>			
First Name	Last Name	Email	Company
Dennis	Gordeev	dennis gordeev@hotmail.com	ABC
Dennis	Gordeev	dennis gordeev@hotmail.com	ABC
Dennis	Gordeev	dennis gordeev@hotmail.com	ABC
Dennis	Gordeev	dennis gordeev@hotmail.com	ABC