ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Старший преподаватель |  |  |  | Соловьева Н.А. |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №7  Язык HTML. Приемы верстки |
| **по дисциплине: Web-Технологии** |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. | 4134К |  |  |  | Самарин Д. В. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург

2023

Оглавление

[Цель работы: 3](#_Toc167854813)

[Вариант задания: 3](#_Toc167854814)

[Описание выполненных пунктов базового и расширенного заданий: 3](#_Toc167854815)

Цель работы: изучение и применение на практике технологии AJAX.

## Вариант задания:

|  |  |
| --- | --- |
| Тип запроса | Вариант задания |
| Post | 4 |

|  |  |
| --- | --- |
| Тема: | Система охлаждения компьютеров |

## Описание выполненных пунктов базового и расширенного заданий:

Базовое:

Разработать AJAX-приложение для доступа к базе данных, подготовленной в предыдущей лабораторной работе. Вариант задания и тип запроса указаны в таблице 1. Формулировку варианта адаптировать под свою базу данных. В сценарии javascript запрещается использование библиотек. Новая страница должна быть встроена в сайт, подготовленный в предыдущих разделах.

Расширенное:

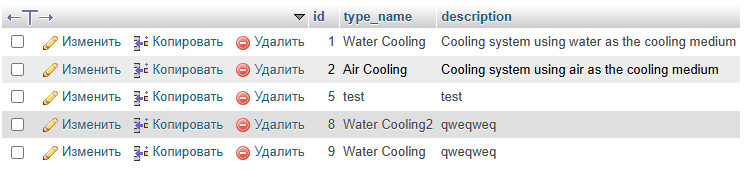
1. использовать json
2. проверить работу в разных браузерах. В сценарии на javascript добавить вывод на страницу названия объекта, выполняющего асинхронный обмен, и название браузера. В отчете привести доказательства этой работы.

Рассмотрим структуру таблиц БД:

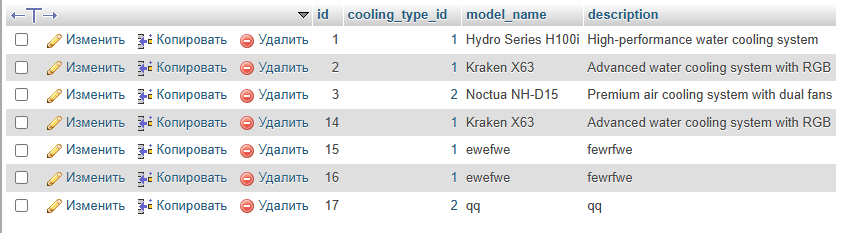
Таблицы связаны между собой отношением "один ко многим", где одна запись в таблице cooling\_types может быть связана с несколькими записями в таблице models.

Рассмотрим данные, хранимые в таблицах:

Содержимое таблицы типы систем охлаждения:



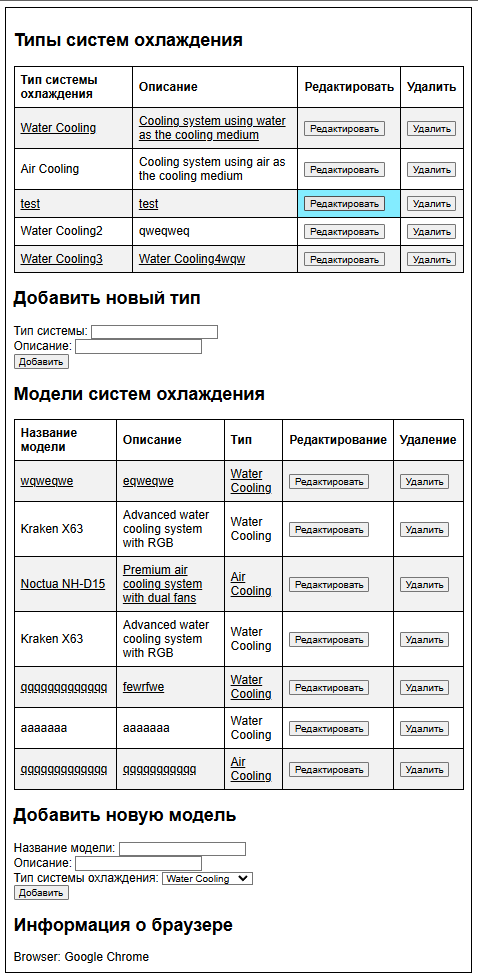
Содержимое таблицы модели систем охлаждения:



Листинг:

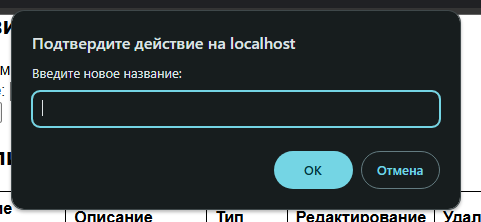
|  |
| --- |
| Index.php  <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <title>Cooling Types and Models with AJAX</title>      <script src="script.js" defer></script>  </head>  <body>      <h2>Типы систем охлаждения</h2>      <table id="coolingTypesTable" border="1">          <tr><th>Тип системы охлаждения</th><th>Описание</th><th>Редактировать</th><th>Удалить</th></tr>      </table>      <h2>Добавить новый тип</h2>      <form id="addCoolingTypeForm">          Тип системы: <input type="text" id="newCoolingTypeName"><br>          Описание: <input type="text" id="newCoolingTypeDescription"><br>          <button type="button" onclick="addCoolingType()">Добавить</button>      </form>      <h2>Модели систем охлаждения</h2>      <table id="modelsTable" border="1">          <tr><th>Название модели</th><th>Описание</th><th>Тип</th><th>Редактирование</th><th>Удаление</th></tr>      </table>      <h2>Добавить новую модель</h2>      <form id="addModelForm">          Название модели: <input type="text" id="newModelName"><br>          Описание: <input type="text" id="newModelDescription"><br>          Тип системы охлаждения:          <select id="newCoolingTypeId"></select><br>          <button type="button" onclick="addModel()">Добавить</button>      </form>      <h2>Информация о браузере</h2>      <div id="browserInfo"></div>      <!-- Добавленный блок для вывода предупреждений -->      <div id="errorMessage" style="color: red; font-weight: bold;"></div>  </body>  </html>  script.js  document.addEventListener('DOMContentLoaded', function() {      loadCoolingTypes();      loadModels();      displayBrowserInfo();  });  function createXHR() {      let xhr = new XMLHttpRequest();      xhr.onreadystatechange = function() {          if (xhr.readyState === XMLHttpRequest.OPENED) {              console.log('AJAX Object: XMLHttpRequest');          }      };      return xhr;  }  function loadCoolingTypes() {      let xhr = createXHR();      xhr.open('POST', 'ajax.php', true);      xhr.setRequestHeader('Content-Type', 'application/json');      xhr.onload = function() {          if (this.status == 200) {              let coolingTypes = JSON.parse(this.responseText);              let output = '<tr><th>Тип системы охлаждения</th><th>Описание</th><th>Редактировать</th><th>Удалить</th></tr>';              coolingTypes.forEach(function(type) {                  output += `<tr>                      <td>${type.type\_name}</td>                      <td>${type.description}</td>                      <td><button onclick="editCoolingType(${type.id})">Редактировать</button></td>                      <td><button onclick="deleteCoolingType(${type.id})">Удалить</button></td>                  </tr>`;              });              document.getElementById('coolingTypesTable').innerHTML = output;              // Populate select dropdown for models              let select = document.getElementById('newCoolingTypeId');              select.innerHTML = ''; // Clear existing options              coolingTypes.forEach(function(type) {                  let option = document.createElement('option');                  option.value = type.id;                  option.text = type.type\_name;                  select.add(option);              });          }      };      xhr.send(JSON.stringify({action: 'getCoolingTypes'}));  }  function loadModels() {      let xhr = createXHR();      xhr.open('POST', 'ajax.php', true);      xhr.setRequestHeader('Content-Type', 'application/json');      xhr.onload = function() {          if (this.status == 200) {              let models = JSON.parse(this.responseText);              let output = '<tr><th>Название модели</th><th>Описание</th><th>Тип</th><th>Редактирование</th><th>Удаление</th></tr>';              models.forEach(function(model) {                  output += `<tr>                      <td>${model.model\_name}</td>                      <td>${model.description}</td>                      <td>${model.type\_name}</td>                      <td><button onclick="editModel(${model.id})">Редактировать</button></td>                      <td><button onclick="deleteModel(${model.id})">Удалить</button></td>                  </tr>`;              });              document.getElementById('modelsTable').innerHTML = output;          }      };      xhr.send(JSON.stringify({action: 'getModels'}));  }  function addCoolingType() {      let typeName = document.getElementById('newCoolingTypeName').value;      let description = document.getElementById('newCoolingTypeDescription').value;      let xhr = createXHR();      xhr.open('POST', 'ajax.php', true);      xhr.setRequestHeader('Content-Type', 'application/json');      xhr.onload = function() {          if (this.status == 200) {              loadCoolingTypes();          } else {              let responseData = JSON.parse(this.responseText);              if (responseData.error) {                  displayErrorMessage(responseData.error); // Display error message              }          }      };      xhr.onerror = function() {          displayErrorMessage('Failed to add cooling type. Please try again.'); // Display error message      };      xhr.send(JSON.stringify({          action: 'addCoolingType',          type\_name: typeName,          description: description      }));  }  function displayErrorMessage(message) {      document.getElementById('errorMessage').innerText = message; // Display error message  }  function addModel() {      let modelName = document.getElementById('newModelName').value;      let description = document.getElementById('newModelDescription').value;      let coolingTypeId = document.getElementById('newCoolingTypeId').value;      let xhr = createXHR();      xhr.open('POST', 'ajax.php', true);      xhr.setRequestHeader('Content-Type', 'application/json');      xhr.onload = function() {          if (this.status == 200) {              loadModels();          }      };      xhr.send(JSON.stringify({          action: 'addModel',          model\_name: modelName,          description: description,          cooling\_type\_id: coolingTypeId      }));  }  function editCoolingType(id) {      let newName = prompt("Enter new name:");      let newDescription = prompt("Enter new description:");      if (newName && newDescription) {          let xhr = createXHR();          xhr.open('POST', 'ajax.php', true);          xhr.setRequestHeader('Content-Type', 'application/json');          xhr.onload = function() {              if (this.status == 200) {                  loadCoolingTypes();              }          };          xhr.send(JSON.stringify({              action: 'updateCoolingType',              id: id,              type\_name: newName,              description: newDescription          }));      }  }  function editModel(id) {      let newName = prompt("Enter new name:");      let newDescription = prompt("Enter new description:");      let newCoolingTypeId = prompt("Enter new cooling type ID:");      if (newName && newDescription && newCoolingTypeId) {          let xhr = createXHR();          xhr.open('POST', 'ajax.php', true);          xhr.setRequestHeader('Content-Type', 'application/json');          xhr.onload = function() {              if (this.status == 200) {                  loadModels();              }          };          xhr.send(JSON.stringify({              action: 'updateModel',              id: id,              model\_name: newName,              description: newDescription,              cooling\_type\_id: newCoolingTypeId          }));      }  }  function deleteCoolingType(id) {      if (confirm("Are you sure you want to delete this cooling type?")) {          let xhr = createXHR();          xhr.open('POST', 'ajax.php', true);          xhr.setRequestHeader('Content-Type', 'application/json');          xhr.onload = function() {              if (this.status == 200) {                  loadCoolingTypes();              }          };          xhr.send(JSON.stringify({action: 'deleteCoolingType', id: id}));      }  }  function deleteModel(id) {      if (confirm("Are you sure you want to delete this model?")) {          let xhr = createXHR();          xhr.open('POST', 'ajax.php', true);          xhr.setRequestHeader('Content-Type', 'application/json');          xhr.onload = function() {              if (this.status == 200) {                  loadModels();              }          };          xhr.send(JSON.stringify({action: 'deleteModel', id: id}));      }  }  function displayBrowserInfo() {      let browser = getBrowserName();      let browserInfo = `Browser: ${browser}`;      document.getElementById('browserInfo').innerText = browserInfo;  }  function getBrowserName() {      let userAgent = navigator.userAgent;      let browserName;      if (userAgent.includes("Chrome")) {          browserName = "Google Chrome";      } else if (userAgent.includes("Firefox")) {          browserName = "Mozilla Firefox";      } else if (userAgent.includes("Safari")) {          browserName = "Safari";      } else if (userAgent.includes("Edge")) {          browserName = "Microsoft Edge";      } else if (userAgent.includes("Opera") || userAgent.includes("OPR")) {          browserName = "Opera";      } else {          browserName = "Unknown";      }      return browserName;  }  ajax.php  <?php  $servername = "localhost";  $username = "root";  $password = "";  $dbname = "laba6";  $conn = new mysqli($servername, $username, $password, $dbname);  if ($conn->connect\_error) {      die("Connection failed: " . $conn->connect\_error);  }  header('Content-Type: application/json');  if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {      $input = json\_decode(file\_get\_contents('php://input'), true);      $action = $input['action'] ?? '';      if ($action == 'getCoolingTypes') {          $sql = "SELECT \* FROM cooling\_types";          $result = $conn->query($sql);          $data = [];          if ($result) {              while ($row = $result->fetch\_assoc()) {                  $data[] = $row;              }          }          echo json\_encode($data);      } elseif ($action == 'getModels') {          $sql = "SELECT models.id, models.model\_name, models.description, cooling\_types.type\_name                  FROM models                  INNER JOIN cooling\_types ON models.cooling\_type\_id = cooling\_types.id";          $result = $conn->query($sql);          $data = [];          if ($result) {              while ($row = $result->fetch\_assoc()) {                  $data[] = $row;              }          }          echo json\_encode($data);      } elseif ($action == 'addCoolingType') {          $type\_name = $input['type\_name'] ?? '';          $description = $input['description'] ?? '';          // Check if cooling type already exists          $stmt = $conn->prepare("SELECT COUNT(\*) as count FROM cooling\_types WHERE type\_name=?");          $stmt->bind\_param('s', $type\_name);          $stmt->execute();          $result = $stmt->get\_result();          $row = $result->fetch\_assoc();          if ($row['count'] > 0) {              echo json\_encode(['error' => 'Cooling type already exists']);          } else {              $stmt = $conn->prepare("INSERT INTO cooling\_types (type\_name, description) VALUES (?, ?)");              $stmt->bind\_param('ss', $type\_name, $description);              $stmt->execute();              if ($stmt->affected\_rows > 0) {                  echo json\_encode(['message' => 'Cooling type added successfully']);              } else {                  echo json\_encode(['error' => 'Failed to add cooling type']);              }          }          $stmt->close();      } elseif ($action == 'addModel') {          $model\_name = $input['model\_name'] ?? '';          $description = $input['description'] ?? '';          $cooling\_type\_id = $input['cooling\_type\_id'] ?? 0;          // Check if model already exists          $stmt = $conn->prepare("SELECT COUNT(\*) as count FROM models WHERE model\_name=?");          $stmt->bind\_param('s', $model\_name);          $stmt->execute();          $result = $stmt->get\_result();          $row = $result->fetch\_assoc();          if ($row['count'] > 0) {              echo json\_encode(['error' => 'Model already exists']);          } else {              $stmt = $conn->prepare("INSERT INTO models (model\_name, description, cooling\_type\_id) VALUES (?, ?, ?)");              $stmt->bind\_param('ssi', $model\_name, $description, $cooling\_type\_id);              $stmt->execute();              if ($stmt->affected\_rows > 0) {                  echo json\_encode(['message' => 'Model added successfully']);              } else {                  echo json\_encode(['error' => 'Failed to add model']);              }          }          $stmt->close();      } elseif ($action == 'deleteCoolingType') {          $id = $input['id'] ?? 0;          $stmt = $conn->prepare("DELETE FROM cooling\_types WHERE id=?");          $stmt->bind\_param('i', $id);          $stmt->execute();          if ($stmt->affected\_rows > 0) {              echo json\_encode(['message' => 'Cooling type deleted successfully']);          } else {              echo json\_encode(['error' => 'Failed to delete cooling type']);          }          $stmt->close();      } elseif ($action == 'deleteModel') {          $id = $input['id'] ?? 0;          $stmt = $conn->prepare("DELETE FROM models WHERE id=?");          $stmt->bind\_param('i', $id);          $stmt->execute();          if ($stmt->affected\_rows > 0) {              echo json\_encode(['message' => 'Model deleted successfully']);          } else {              echo json\_encode(['error' => 'Failed to delete model']);          }          $stmt->close();      } elseif ($action == 'updateCoolingType') {          $id = $input['id'] ?? 0;          $type\_name = $input['type\_name'] ?? '';          $description = $input['description'] ?? '';          $stmt = $conn->prepare("UPDATE cooling\_types SET type\_name=?, description=? WHERE id=?");          $stmt->bind\_param('ssi', $type\_name, $description, $id);          $stmt->execute();          if ($stmt->affected\_rows > 0) {              echo json\_encode(['message' => 'Cooling type updated successfully']);          } else {              echo json\_encode(['error' => 'Failed to update cooling type']);          }          $stmt->close();      } elseif ($action == 'updateModel') {          $id = $input['id'] ?? 0;          $model\_name = $input['model\_name'] ?? '';          $description = $input['description'] ?? '';          $cooling\_type\_id = $input['cooling\_type\_id'] ?? 0;          $stmt = $conn->prepare("UPDATE models SET model\_name=?, description=?, cooling\_type\_id=? WHERE id=?");          $stmt->bind\_param('ssii', $model\_name, $description, $cooling\_type\_id, $id);          $stmt->execute();          if ($stmt->affected\_rows > 0) {              echo json\_encode(['message' => 'Model updated successfully']);          } else {              echo json\_encode(['error' => 'Failed to update model']);          }          $stmt->close();      } else {          echo json\_encode(['error' => 'Invalid action']);      }  }  $conn->close();  ?> |

Рассмотрим результат выполненной работы:

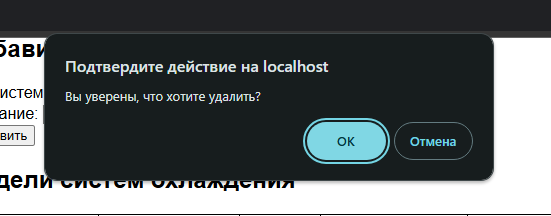


Присутствует удаление/редактирование/добавление записей.

Редактирование:



Удаление:



Вывод: в ходе лабораторной работы изучены и применены на практике технологии AJAX.