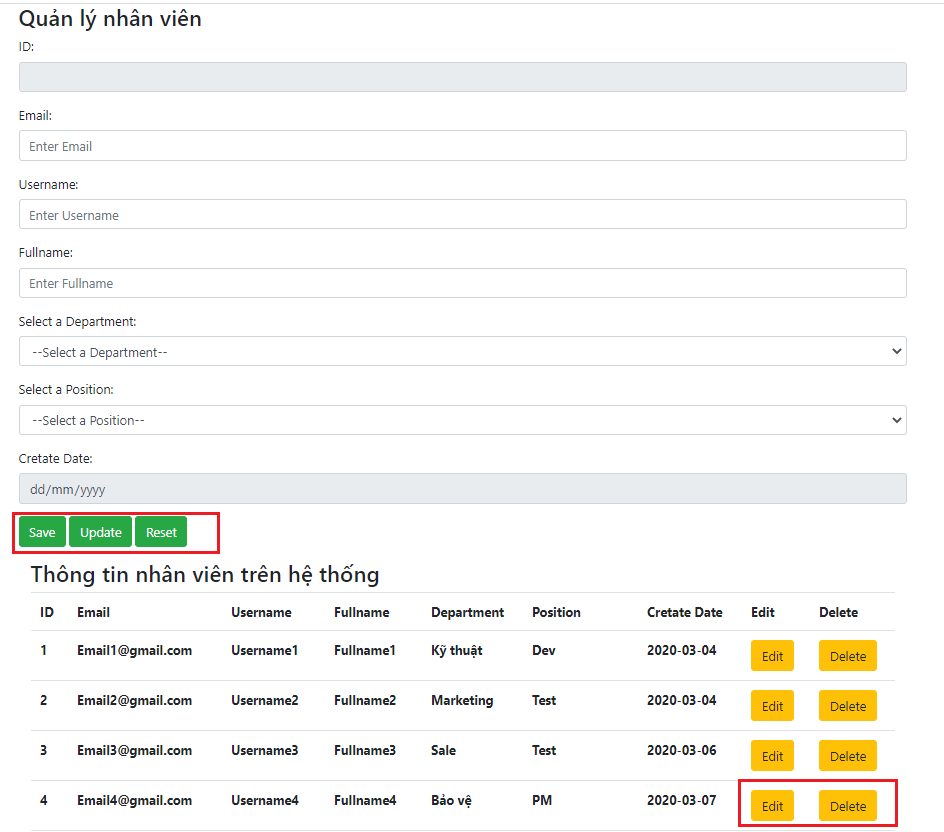
# LAB\_8 Tích hợp Frontend cho LAB\_3\_Chức năng CRUD cơ bản:

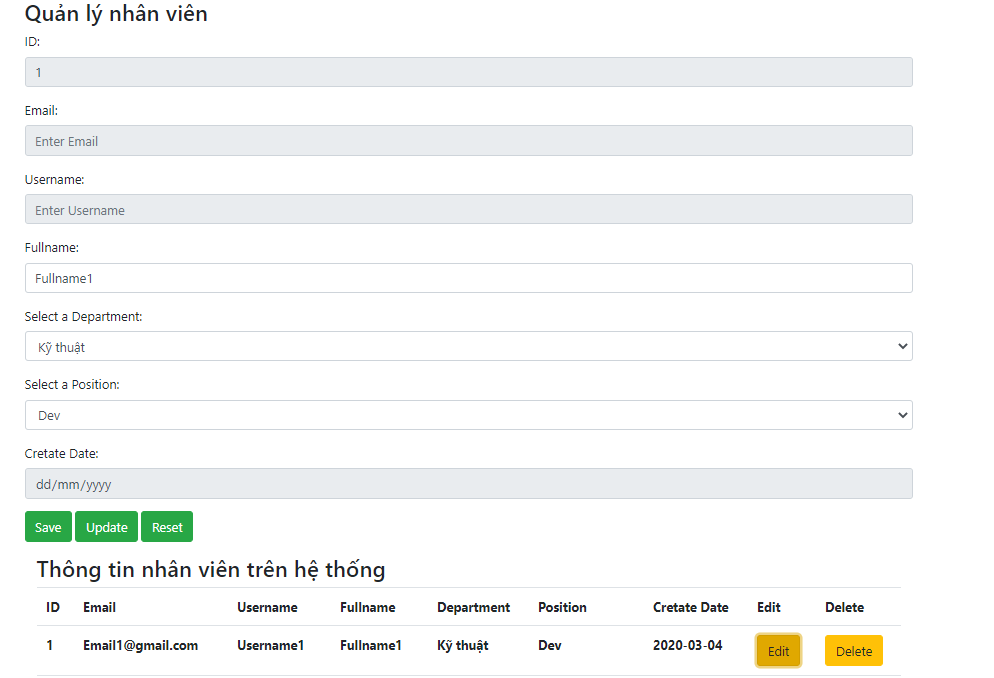
* Sử dụng lại phần tạo API trong LAB 3 từ Spring Boot Project, tích hợp thêm giao diện cho người sử dụng.



# Giao diện chính chương trình :



* Giao diện Update:



* Thực hiện các chức năng xem, thêm xóa sửa theo các nút chức năng tương ứng.

# Tạo trang html

* Trang này sử dụng lại phần Lab get dữ liệu API đã làm trong phần học Frontend: chú ý comment lại các thẻ Select- Option cho dữ liệu Department và Possition, các dữ liệu này sẽ lấy từ Server API và gen qua Jquery.

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta http-equiv="X-UA-Compatible" content="IE=edge">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>Quản lý nhân viên With AJAX </title>      <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">      <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>      <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>      <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>      <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.4.0/font/bootstrap-icons.css">      <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>  </head>  <body>      <div class="container">          <div class="form">              <h3>Quản lý nhân viên</h3>              <form method="POST" id="Main\_Form\_ID">                  <div class="form-group">                      <label for="">ID: </label>                      <input type="number" required="true" class="form-control" id="ID\_ID" value="">                  </div>                  <div class="form-group">                      <label for="">Email: </label>                      <input type="email" required="true" class="form-control" id="Email\_ID" value="" placeholder="Enter Email">                  </div>                  <div class="form-group">                      <label for="">Username: </label>                      <input type="text" required="true" class="form-control" id="Username\_ID" value="" placeholder="Enter Username">                  </div>                  <div class="form-group">                      <label for="">Fullname: </label>                      <input type="text" required="true" class="form-control" id="Fullname\_ID" value="" placeholder="Enter Fullname">                  </div>                  <div class="form-group">                      <label for="sel1">Select a Department:</label>                      <select class="form-control" id="Department\_ID">                    <option>--Select a Department--</option>                   <!--  Comment đoạn code này lại, sẽ thực hiện load và fill dữ liệu API lấy trực tiếp từ Server                     <option>Accounting</option>                     <option>Business Development</option>                    <option>Engineering</option>                    <option>Human Resources</option>                    <option>Research and Development</option>   -->                  </select>                  </div>                  <div class="form-group">                      <label for="sel1">Select a Position:</label>                      <select class="form-control" id="Position\_ID">                     <option>--Select a Position--</option>                   <!-- Comment đoạn code này lại, sẽ thực hiện load và fill dữ liệu API lấy trực tiếp từ Server                    <option>Developer</option>                    <option>Tester</option>                    <option>Scrum Master</option>                    <option>PM</option> -->                  </select>                  </div>                  <div class="form-group">                      <label for="">Cretate Date: </label>                      <input type="date" required="true" class="form-control" id="Cretate\_Date\_ID" value="">                  </div>                  <div class="form-group">                      <input type="submit" class="btn btn-success" value="Save">                      <input type="button" class="btn btn-success" id="update\_btn" value="Update">                      <input type="button" class="btn btn-success" id="reset\_btn" value="Reset">                  </div>              </form>          </div>          <div class="container">              <div class="form">                  <h3>Thông tin nhân viên trên hệ thống</h3>                  <table class="table table-hover" id="Detail\_Staff\_ID">                      <thead>                          <tr>                              <th>ID</th>                              <th>Email</th>                              <th>Username</th>                              <th>Fullname</th>                              <th>Department</th>                              <th>Position</th>                              <th>Cretate Date</th>                              <th>Edit</th>                              <th>Delete</th>                          </tr>                      </thead>                      <tbody id="Result\_TB">                      </tbody>                  </table>              </div>          </div>      </div>      <script type="text/javascript" src="QLNV\_DNQ\_Script.js"></script>  </body>  </html> |

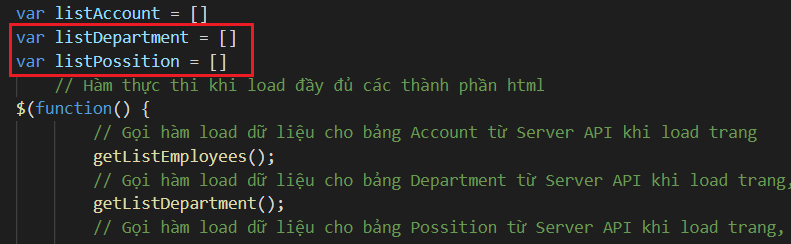
# Tạo trang js để xử lý:

* 1. Fill dữ liệu API vào các Dropdown List: Department và Possition.
* Comment lại các đoạn Code html liên quan tới HardCode bên html của Department và Possition.



|  |
| --- |
| <div class="form-group">                      <label for="sel1">Select a Department:</label>                      <select class="form-control" id="Department\_ID">                    <option>--Select a Department--</option>                    <!-- <option>Accounting</option>                    <option>Business Development</option>                    <option>Engineering</option>                    <option>Human Resources</option>                    <option>Research and Development</option> -->                  </select>                  </div>                  <div class="form-group">                      <label for="sel1">Select a Position:</label>                      <select class="form-control" id="Position\_ID">                    <option>--Select a Position--</option>                    <!-- <option>Developer</option>                    <option>Tester</option>                    <option>Scrum Master</option>                    <option>PM</option> -->                  </select>                  </div> |

* Tạo thêm 2 mảng để lưu thông tin Department và Possition tương ứng.



|  |
| --- |
| var listDepartment = []  var listPossition = [] |

* Tạo thêm các hàm: getListDepartment(); getListPossition(); để lấy dữ liệu cho Department và Possition.



|  |
| --- |
| function getListDepartment() {      // call API from server      $.get("http://localhost:8080/api/v1/departments/", function(data, status) {          // reset list employees          listDepartment = [];          // error          if (status == "error") {              // TODO              alert("Error when loading data");              return;          }          // success thì sẽ đổ dữ liệu vào ListDepartment và lấy dữ liệu trong List này fill vào phần thẻ Selecr để chọn Department, chú ý phần này phải để trong nội dung xử lý của Ajax          data.forEach(function(item) {              var department = {                  'id': item.id,                  'name': item.name,              }              listDepartment.push(department)          });          for (let index = 0; index < listDepartment.length; index++) {              $('#Department\_ID').append(`          <option>${listDepartment[index].name}</option>            `)          }      });  }  function getListPossition() {      // call API from server      $.get("http://localhost:8080/api/v1/possitions/", function(data, status) {          // reset list employees          listPossition = [];          // error          if (status == "error") {              // TODO              alert("Error when loading data");              return;          }          // success thì sẽ đổ dữ liệu vào listPossition và lấy dữ liệu trong List này fill vào phần thẻ Selecr để chọn Possition, chú ý phần này phải để trong nội dung xử lý của Ajax          data.forEach(function(item) {              var possition = {                  'id': item.id,                  'name': item.name,              }              listPossition.push(possition)          });          for (let index = 0; index < listPossition.length; index++) {              $('#Position\_ID').append(`          <option>${listPossition[index].name}</option>            `)          }      });  } |

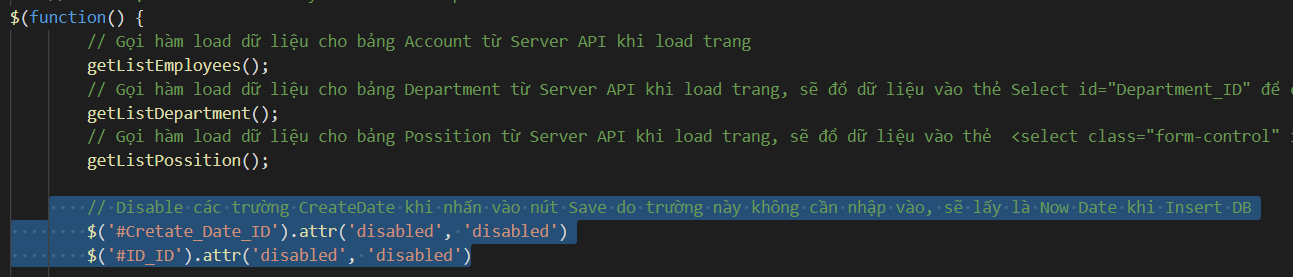
* Thêm các hàm getListDepartment(); getListPossition(); vào đầu chương trình để khai báo gọi các hàm này khi chương trình load lên, đổ dữ liệu vào các DropdownList tương ứng.
  1. Thực hiện sửa lại chức năng Show list Account.
* Sửa lại hàm: getListEmployees(); thay thế link API tương ứng Server backend đẩy ra: **http://localhost:8080/api/v1/accounts/**

|  |
| --- |
| function getListEmployees() {      // call API from server      $.get("http://localhost:8080/api/v1/accounts/", function(data, status) {          // reset list employees          listAccount = [];          // error          if (status == "error") {              // TODO              alert("Error when loading data");              return;          }          // success          parseData(data);          showAccount();      });  } |

* Sửa lại hàm parseData(data), các thuộc tính tương ứng với AccountDTO:

|  |
| --- |
| function parseData(data) {      // employees = data;      data.forEach(function(item) {          var account = {              'AccountID': item.id,              'Email': item.email,              'Username': item.username,              'FullName': item.fullname,              'Department': item.department,              'Position': item.position,              'CreateDate': item.createDate,          }          listAccount.push(account)      });  } |

* 1. Thực hiện sửa lại chức năng thêm mới Account.
* Cập nhật lại sự kiện khi nhấn nút Save (Submit), khi thêm mới Account thì không cần 2 trường Id, Create\_Date nên vì Id đã tự động tăng và Create\_Date lấy theo ngày giờ hệ thống. nên sẽ Disable 2 trường này trong html để người dùng không nhập vào.



|  |
| --- |
| // Disable các trường CreateDate khi nhấn vào nút Save do trường này không cần nhập vào, sẽ lấy là Now Date khi Insert DB          $('#Cretate\_Date\_ID').attr('disabled', 'disabled')          $('#ID\_ID').attr('disabled', 'disabled') |

* Viết thêm 2 đoạn code để lấy được DepID và PosID từ lựa chọn tên Department và Possition từ người dùng chọn trên DropdownList.

|  |
| --- |
| // Lấy ra ID của Department khi người dùng lựa chọn phòng ban              for (let index = 0; index < listDepartment.length; index++) {                  if (listDepartment[index].name == v\_Department\_ID) {                      var depID = listDepartment[index].id                  }              }              // Lấy ra ID của Possition khi người dùng lựa chọn Possition              for (let index = 0; index < listPossition.length; index++) {                  if (listPossition[index].name == v\_Position\_ID) {                      var posID = listPossition[index].id                  }              } |

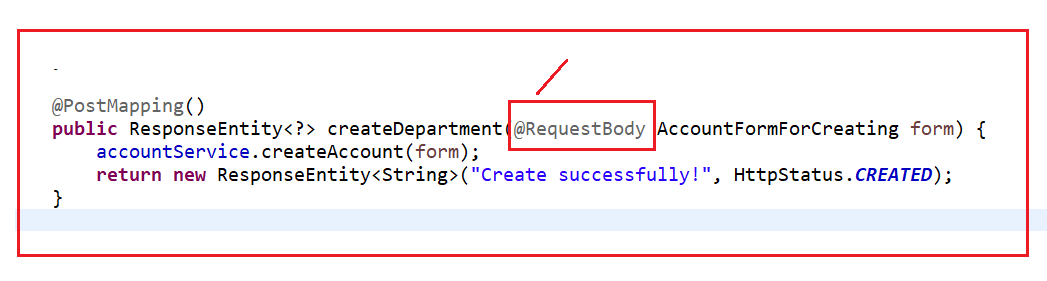
* Sửa lại đối tượng Account chứa thông tin gửi đi sao cho các thuộc tính tương ứng với **AccountFormForCreating** bên Backend. Comment lại trường **'CreateDate'** do đã được lấy tự động bằng thời gian hệ thống

|  |
| --- |
| var account = {                      'email': v\_Email\_ID,                      'username': v\_Username\_ID,                      'fullname': v\_Fullname\_ID,                      'departmentId': depID,                      'positionId': posID,                      // 'CreateDate': v\_Cretate\_Date\_ID                  } |

* Sửa lại đường link Backend tương ứng khi gọi Ajax.

|  |
| --- |
| $.post("http://localhost:8080/api/v1/accounts/", account,                  function(data, status) {                      // error                      if (status == "error") {                          alert("Error when loading data");                          return;                      }                      // success                      // Hàm này để hiển thị thông tin account ở table                      getListEmployees();                  });              return false;              // Sử dụng return false để không redirect tới 1 trang khác. |

* Khi thực hiện demo bỏ @RequestBody trong @PostMapping()



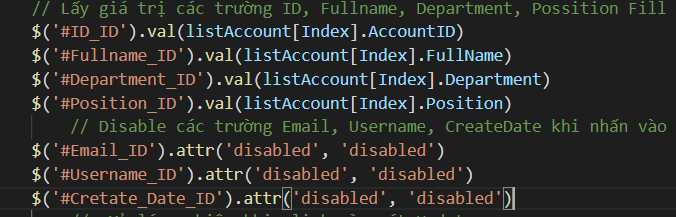
* 1. Thực hiện sửa lại chức năng Update Account.
* Trong bài tập này chỉ thực hiện Update các trường: FullName, Department, Position. Nên sẽ thực hiện xóa bớt các trường không cần lấy dữ liệu:

|  |
| --- |
| $('#ID\_ID').val(listAccount[Index].AccountID)      $('#Email\_ID').val(listAccount[Index].Email)      $('#Username\_ID').val(listAccount[Index].Username)      $('#Fullname\_ID').val(listAccount[Index].FullName)      $('#Department\_ID').val(listAccount[Index].Department)      $('#Position\_ID').val(listAccount[Index].Position)      $('#Cretate\_Date\_ID').val(listAccount[Index].CreateDate) |

🡺 Kết quả:

|  |
| --- |
| var v\_ID\_ID = $('#ID\_ID').val()          var v\_Fullname\_ID = $('#Fullname\_ID').val()          var v\_Fullname\_ID = $('#Fullname\_ID').val()          var v\_Department\_ID = $('#Department\_ID').val()          var v\_Position\_ID = $('#Position\_ID').val() |

* Thực hiện Disable các trường không cho Update:

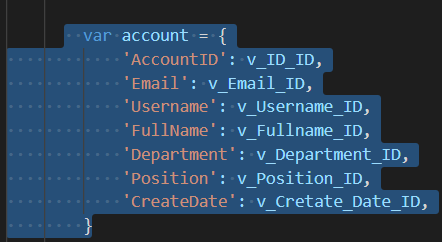


|  |
| --- |
| $('#Email\_ID').attr('disabled', 'disabled')      $('#Username\_ID').attr('disabled', 'disabled')      $('#Cretate\_Date\_ID').attr('disabled', 'disabled') |

* Viết thêm các đoạn code lấy DepID và PosID:

|  |
| --- |
| // Lấy ra ID của Department khi người dùng lựa chọn phòng ban          for (let index = 0; index < listDepartment.length; index++) {              if (listDepartment[index].name == v\_Department\_ID) {                  var depID = listDepartment[index].id              }          }          // Lấy ra ID của Possition khi người dùng lựa chọn Possition          for (let index = 0; index < listPossition.length; index++) {              if (listPossition[index].name == v\_Position\_ID) {                  var posID = listPossition[index].id              }          } |

* Sửa lại thông tin Account cần gửi đi để Update:



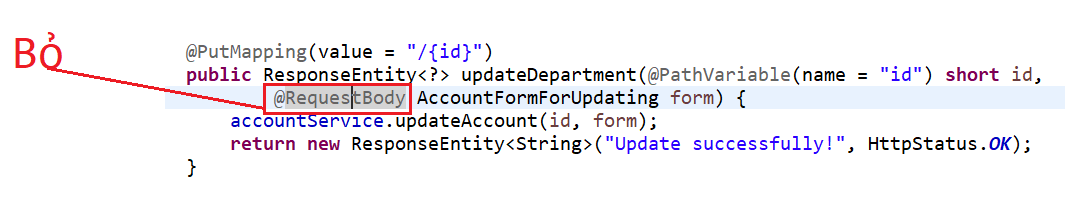
🡺Sửa lại:

|  |
| --- |
| var account = {              'fullname': v\_Fullname\_ID,              'departmentId': depID,              'positionId': posID,          } |

* Thay đổi lại đường Link API qua Ajax tương ứng.

|  |
| --- |
| $.ajax({              url: 'http://localhost:8080/api/v1/accounts/' + v\_ID\_ID,              type: 'PUT',              data: account,              success: function(result) {                  // error                  if (result == undefined || result == null) {                      alert("Error when loading data");                      return;                  }                  getListEmployees();              }          }); |

* Khi thực hiện demo bỏ @RequestBody trong @PutMapping(value = "/{id}")



* 1. Thực hiện sửa lại chức năng Xóa Account.
* Phần này chỉ cần sửa lại đường link API Backend tương ứng.

|  |
| --- |
| if (confirm\_del) {          $.ajax({              url: 'http://localhost:8080/api/v1/accounts/' + v\_del\_ID,              type: 'DELETE',              success: function(result) {                  // error                  if (result == undefined || result == null) {                      alert("Error when loading data");                      return;                  }                  // success                  getListEmployees();              }          });      } else {          return      } |

* 1. Đoạn Code hoàn chỉnh

|  |
| --- |
| // Trong bài tập này sẽ xử dụng Jquery và JavaScript, JavaScript để thực hiện chức năng quản lý nhân viên, tương ứng với bảng Account trong DB TestingSystem, Các chức năng: Thêm, Sửa, Xóa. Dữ liệu này sẽ tương tác với server Backend API.  // Khai báo 1 array để lưu thông tin tất cả các Account trên hệ thống.  var listAccount = []  var listDepartment = []  var listPossition = []      // Hàm thực thi khi load đầy đủ các thành phần html  $(function() {          // Gọi hàm load dữ liệu cho bảng Account từ Server API khi load trang          getListEmployees();          // Gọi hàm load dữ liệu cho bảng Department từ Server API khi load trang, sẽ đổ dữ liệu vào thẻ Select id="Department\_ID" để chọn phòng ban          getListDepartment();          // Gọi hàm load dữ liệu cho bảng Possition từ Server API khi load trang, sẽ đổ dữ liệu vào thẻ  <select class="form-control" id="Position\_ID"> để chọn Possition cho Account          getListPossition();          // Disable các trường CreateDate khi nhấn vào nút Save do trường này không cần nhập vào, sẽ lấy là Now Date khi Insert DB          $('#Cretate\_Date\_ID').attr('disabled', 'disabled')          $('#ID\_ID').attr('disabled', 'disabled')          // Xử lý sự kiện cho nút reset, Sử dụng JQuery để lấy các giá trị các trường theo ID, sau đó Set về rỗng          $('#reset\_btn').click(function() {              $('#ID\_ID').val("")              $('#Email\_ID').val("")              $('#Username\_ID').val("")              $('#Fullname\_ID').val("")              $('#Department\_ID').val("")              $('#Position\_ID').val("")              $('#Cretate\_Date\_ID').val("")          })          // Xử lý sự kiện khi nhấn nút Save          // Xử lý sự kiện khi nhấn nút Submit của Form(Save), ở đây phải sử dụng Return False ở cuối sự kiện này để không redirect sang trang mới.          $('#Main\_Form\_ID').submit(function() {              // Lấy các giá trị người dùng nhập vào              var v\_ID\_ID = $('#ID\_ID').val()              var v\_Email\_ID = $('#Email\_ID').val()              var v\_Username\_ID = $('#Username\_ID').val()              var v\_Fullname\_ID = $('#Fullname\_ID').val()              var v\_Department\_ID = $('#Department\_ID').val()              var v\_Position\_ID = $('#Position\_ID').val()                  // var v\_Cretate\_Date\_ID = $('#Cretate\_Date\_ID').val()                  // Lấy ra ID của Department khi người dùng lựa chọn phòng ban              for (let index = 0; index < listDepartment.length; index++) {                  if (listDepartment[index].name == v\_Department\_ID) {                      var depID = listDepartment[index].id                  }              }              // Lấy ra ID của Possition khi người dùng lựa chọn Possition              for (let index = 0; index < listPossition.length; index++) {                  if (listPossition[index].name == v\_Position\_ID) {                      var posID = listPossition[index].id                  }              }              // Tạo 1 đối tượng account để lưu giữ thông tin nhận được, ở đây khi tạo đối tượng không cần lấy ID do khi tạo dữ liệu trên server ID sẽ được tự động Gen              var account = {                      'email': v\_Email\_ID,                      'username': v\_Username\_ID,                      'fullname': v\_Fullname\_ID,                      'departmentId': depID,                      'positionId': posID,                      // 'CreateDate': v\_Cretate\_Date\_ID                  }                  // Add account tới MOCK API              $.post("http://localhost:8080/api/v1/accounts/", account,                  function(data, status) {                      // error                      if (status == "error") {                          alert("Error when loading data");                          return;                      }                      // success                      // Hàm này để hiển thị thông tin account ở table                      getListEmployees();                  });              return false;              // Sử dụng return false để không redirect tới 1 trang khác.          })      })      // Đoạn bên trên khi load hoàn chỉnh trang web sẽ kết thúc ở đây, đoạn dưới sẽ viết các hàm bổ trợ khác      // Viết hàm showAccount()  function showAccount() {      $('#Email\_ID').removeAttr("disabled")      $('#Username\_ID').removeAttr("disabled")          // Xóa hết kết quả đang hiển thị ở bảng kết quả      $('#Result\_TB').empty()          // Lặp trong listAccount để in thông tin từng phần tử          // Hiển thị thêm 2 nút để sửa và xóa các Account      for (var index = 0; index < listAccount.length; index++) {          $('#Result\_TB').append(`                <tr>                <th>${listAccount[index].AccountID}</th>                <th>${listAccount[index].Email}</th>                <th>${listAccount[index].Username}</th>                <th>${listAccount[index].FullName}</th>                <th>${listAccount[index].Department}</th>                <th>${listAccount[index].Position}</th>                <th>${listAccount[index].CreateDate}</th>                <th><button class="btn btn-warning" onclick="editAccount(${index})">Edit</button></th>                <th><button class="btn btn-warning" onclick="deleteAccount(${index})">Delete</button></th>                </tr>                `)      }  }  // Viết hàm xóa account  function deleteAccount(Index) {      var v\_del\_ID = listAccount[Index].AccountID;      // Hiển thị 1 Confim Popup, chọn Có = True      var confirm\_del = confirm('Bạn có chắc chắn muốn xóa Account này không')      if (confirm\_del) {          $.ajax({              url: 'http://localhost:8080/api/v1/accounts/' + v\_del\_ID,              type: 'DELETE',              success: function(result) {                  // error                  if (result == undefined || result == null) {                      alert("Error when loading data");                      return;                  }                  // success                  getListEmployees();              }          });      } else {          return      }  }  // Viết hàm để Edit các account  function editAccount(Index) {      // Lấy giá trị các trường ID, Fullname, Department, Possition Fill vào Form      $('#ID\_ID').val(listAccount[Index].AccountID)      $('#Fullname\_ID').val(listAccount[Index].FullName)      $('#Department\_ID').val(listAccount[Index].Department)      $('#Position\_ID').val(listAccount[Index].Position)          // Disable các trường Email, Username, CreateDate khi nhấn vào nút Edit do không cho cập nhật các trường này      $('#Email\_ID').attr('disabled', 'disabled')      $('#Username\_ID').attr('disabled', 'disabled')      $('#Cretate\_Date\_ID').attr('disabled', 'disabled')          //  Xử lý sự kiện khi click vào nút Update      $('#update\_btn').click(function() {          var v\_ID\_ID = $('#ID\_ID').val()          var v\_Fullname\_ID = $('#Fullname\_ID').val()          var v\_Fullname\_ID = $('#Fullname\_ID').val()          var v\_Department\_ID = $('#Department\_ID').val()          var v\_Position\_ID = $('#Position\_ID').val()              // Lấy ra ID của Department khi người dùng lựa chọn phòng ban          for (let index = 0; index < listDepartment.length; index++) {              if (listDepartment[index].name == v\_Department\_ID) {                  var depID = listDepartment[index].id              }          }          // Lấy ra ID của Possition khi người dùng lựa chọn Possition          for (let index = 0; index < listPossition.length; index++) {              if (listPossition[index].name == v\_Position\_ID) {                  var posID = listPossition[index].id              }          }          // Ở đây chỉ cho người dùng update các trường, fullname, Department, Possition, các trường Username, Email trên thực tế sẽ để cố định.          var account = {              'fullname': v\_Fullname\_ID,              'departmentId': depID,              'positionId': posID,          }          $.ajax({              url: 'http://localhost:8080/api/v1/accounts/' + v\_ID\_ID,              type: 'PUT',              data: account,              success: function(result) {                  // error                  if (result == undefined || result == null) {                      alert("Error when loading data");                      return;                  }                  getListEmployees();              }          });      })  }  // Viết hàm gọi dữ liệu server API bằng Ajax khi load trang Web  function getListEmployees() {      // call API from server      $.get("http://localhost:8080/api/v1/accounts/", function(data, status) {          // reset list employees          listAccount = [];          // error          if (status == "error") {              // TODO              alert("Error when loading data");              return;          }          // success          parseData(data);          showAccount();      });  }  // hàm chuyển dữ liệu API nhận được bằng Ajax chuyển về dạng List để sử dụng trong chương trình  function parseData(data) {      // employees = data;      data.forEach(function(item) {          var account = {              'AccountID': item.id,              'Email': item.email,              'Username': item.username,              'FullName': item.fullname,              'Department': item.department,              'Position': item.position,              'CreateDate': item.createDate,          }          listAccount.push(account)      });  }  function getListDepartment() {      // call API from server      $.get("http://localhost:8080/api/v1/departments/", function(data, status) {          // reset list employees          listDepartment = [];          // error          if (status == "error") {              // TODO              alert("Error when loading data");              return;          }          // success thì sẽ đổ dữ liệu vào ListDepartment và lấy dữ liệu trong List này fill vào phần thẻ Selecr để chọn Department, chú ý phần này phải để trong nội dung xử lý của Ajax          data.forEach(function(item) {              var department = {                  'id': item.id,                  'name': item.name,              }              listDepartment.push(department)          });          for (let index = 0; index < listDepartment.length; index++) {              $('#Department\_ID').append(`          <option>${listDepartment[index].name}</option>            `)          }      });  }  function getListPossition() {      // call API from server      $.get("http://localhost:8080/api/v1/possitions/", function(data, status) {          // reset list employees          listPossition = [];          // error          if (status == "error") {              // TODO              alert("Error when loading data");              return;          }          // success thì sẽ đổ dữ liệu vào listPossition và lấy dữ liệu trong List này fill vào phần thẻ Selecr để chọn Possition, chú ý phần này phải để trong nội dung xử lý của Ajax          data.forEach(function(item) {              var possition = {                  'id': item.id,                  'name': item.name,              }              listPossition.push(possition)          });          for (let index = 0; index < listPossition.length; index++) {              $('#Position\_ID').append(`          <option>${listPossition[index].name}</option>            `)          }      });  } |

# Xử lý phần Backend:

* Nội dung phần Account Controller cần xóa thuộc tính @RequestBody khi để có thể xảy ra các lỗi lặt vặt trong thực thi, chưa tìm ra nguyên nhân.
* Chức năng Delete đang gặp lỗi API phần Backend, khi thực hiện xóa 1 bản ghi thì xóa thêm các bản ghi xung quanh nữa, cần tìm hiểu lại.