



IT3030
Programming Applications and
Frameworks
3rd Year, 1st Semester

Assignment 2020 S1 – Practical Test

Submitted to
Sri Lanka Institute of Information Technology

IT18137156
Aluthgedara C.R.B.
Y3S1.02.1

In partial fulfillment of the requirements for the
Bachelor of Science Special Honors Degree in Information Technology

06.05.2020

GitHub Public Repo –

<https://github.com/ChamelaAluthgedara/HelthCare-PAF-DoctorManagement-PracticalTest.git>

View

- Form with ID.
- Input elements with IDs.

```
<form id="newForm">
  <div class="form-group row">
    <label for="docID" class="col-sm-2 col-form-label">Doctor ID</label>
    <div class="col-sm-10">
      <input type="number" class="form-control" id="docID"
        placeholder="Doctor ID" oninput="maxLengthCheck(this)"
        maxlength="4">
    </div>
  </div>
  <div class="form-group row">
    <label for="docFname" class="col-sm-2 col-form-label">Firstname</label>
    <div class="col-sm-10">
      <input type="text" class="form-control" id="docFname"
        placeholder="Firstname">
    </div>
  </div>
  <div class="form-group row">
    <label for="docLname" class="col-sm-2 col-form-label">Lastname</label>
    <div class="col-sm-10">
      <input type="text" class="form-control" id="docLname"
        placeholder="Lastname">
    </div>
  </div>
</form>
```

```
<div class="form-group row">
  <label for="docPosition" class="col-sm-2 col-form-label">Position</label>
  <div class="col-sm-10">
    <input type="text" class="form-control" id="docPosition"
      placeholder="Position">
  </div>
</div>

<div class="form-group row">
  <label for="docFee" class="col-sm-2 col-form-label">Fee</label>
  <div class="col-sm-10">
    <input type="number" class="form-control" id="docFee"
      placeholder="Fee" oninput="maxLengthCheck(this)" maxlength="10">
  </div>
</div>

<div class="form-group row">
  <label for="docAddress" class="col-sm-2 col-form-label">Address</label>
  <div class="col-sm-10">
    <input type="text" class="form-control" id="docAddress"
      placeholder="Address">
  </div>
</div>

<div class="form-group row">
  <label for="mobileNo" class="col-sm-2 col-form-label">Mobile</label>
  <div class="col-sm-10">
    <input type="number" class="form-control" id="mobileNo"
      placeholder="Mobile" oninput="maxLengthCheck(this)" maxlength="10">
  </div>
</div>

<div class="form-group row">
  <label for="hosID" class="col-sm-2 col-form-label">Hospital
    ID</label>
  <div class="col-sm-10">
    <select id="hosID" name="hosID">
    </select> <label id="editHostId" class="label12 warning"> Edit </label>
  </div>
</div>
</div>
```

- Save button (correct type)
 - Hidden element (for save operation)

```
129●      <div class="btn-group">
130          <button id="submitDoctorBtn" type="submit" class="btn btn-primary">SubmitDetails</button>
131          <button id="updateDoctorBtn" type="submit" class="btn btn-primary">UpdateDetails</button>
132      </div>
133  </form>
134  <hr>
135●  <table id="doctorDetailsTable" class="table table-bordered table-hover">
136●      <thead>
137          <th>Doc ID</th>
138          <th>Doctor Firstname</th>
139          <th>Doctor Lastname</th>
140          <th>Doctor Position</th>
141          <th>Doctor Fee</th>
142          <th>Doctor Address</th>
143          <th>Doctor Mobile</th>
144          <th>Doctor Hospital ID</th>
145      </thead>
146      <tbody id="doctorsDetailssBody"></tbody>
147  </table>
148  </div>
149 </body>
150 </html>
```

- Alerts

- notify.js

```
<script src="notify.js"></script>
```

CSS

```
<link rel="stylesheet"
      href="https://use.fontawesome.com/releases/v5.3.1/css/all.css"
      crossorigin="anonymous">
<link rel="stylesheet"
      href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
<link rel="stylesheet" href="style.css">

<style>
  input::-webkit-outer-spin-button, input::-webkit-inner-spin-button {
    -webkit-appearance: none;
    margin: 0;
  }
  /* Firefox */
  input[type=number] {
    -moz-appearance: textfield;
  }

  .alert {
    width: 350px;
  }

  .btn-group button {
    border: 1px solid black;
    margin-left: 15px;
    /* /* Float the buttons side by side */
  }

  .label12 {
    color: white;
    padding: 8px;
    font-family: Arial;
  }
  .warning {background-color: #ff9800;} /* Orange */

</style>
```

Java Script

```
<script src="https://code.jquery.com/jquery-3.5.0.js"></script>
<script src="script.js"></script>
<script src="notify.js"></script>
```

Form Numbers Length validation

```
<script>
  function maxLengthCheck(object) {
    if (object.value.length > object.maxLength)
      object.value = object.value.slice(0, object.maxLength)
  }
</script>
```

Controller

- Save button click handler (event, selector, function)

```
// ----- Submit Button Click Event -----
$("#submitDoctorBtn").on("click", function(e) {
    let data = {
        docID: $("#newForm")[0].docID.val(),
        docFname: $("#newForm")[0].docFname.val(),
        docLname: $("#newForm")[0].docLname.val(),
        docPosition: $("#newForm")[0].docPosition.val(),
        docFee: $("#newForm")[0].docFee.val(),
        docAddress: $("#newForm")[0].docAddress.val(),
        mobileNo: $("#newForm")[0].mobileNo.val(),
        hosID: $('#hosID :selected').text()
    }
    console.log($('#hosID option:selected').text());

    var status = validateItemForm();

    // If not valid-----
    if (status !== true)
    {
        $.notify(status, "error"); //show error message
    }
    if(status == true){ //field validation
        createDoctor(data);
    }
    $("#newForm").trigger("reset");
    e.preventDefault();

});
//-----
```

- Alerts

```
$.notify(status, "error");
```

- Save button click event handling process.
 - Initialize
 - Use model for validation
 - Show status messages

```
//----- Form Validation -----  
function validateItemForm()  
{  
  
    if ($("#docID").val().trim() == "")  
    {  
        return "Empty Field Detected: Doctor's ID.";  
    }  
    if ($("#docFname").val().trim() == "")  
    {  
        return "Empty Field Detected: Doctor's Firstname.";  
    }  
    if ($("#docLname").val().trim() == "")  
    {  
        return "Empty Field Detected: Doctor's Lastname.";  
    }  
    if ($("#docPosition").val().trim() == "")  
    {  
        return "Empty Field Detected: Doctor's Position.";  
    }  
    if ($("#docFee").val().trim() == "")  
    {  
        return "Empty Field Detected: Doctor's Fee.";  
    }  
    if ($("#docAddress").val().trim() == "")  
    {  
        return "Empty Field Detected: Doctor's Address.";  
    }  
    if ($("#mobileNo").val().trim() == "")  
    {  
        return "Empty Field Detected: Doctor's Mobile Number.";  
    }  
    return true;  
}  
  
//-----
```

```

$(document).ready(function(){

    getRegisteredDoctors();
    $("#updateDoctorBtn").hide();

    function getRegisteredDoctors(){
        $('#doctorsDetailssBody').html('');
        $.ajax({
            url: 'webapi/doctors/',
            method: 'get',
            dataType: 'json',
            success: function(data) {
                $(data).each(function(i, doctor){
                    $('#doctorsDetailssBody').append($("<tr>"
                        .append("<td>").append(doctor.docID))
                        .append("<td>").append(doctor.docFName))
                        .append("<td>").append(doctor.docLName))
                        .append("<td>").append(doctor.docPosition))
                        .append("<td>").append(doctor.docFee))
                        .append("<td>").append(doctor.docAddress))
                        .append("<td>").append(doctor.mobileNo))
                        .append("<td>").append(doctor.hosID))
                        .append("<td>").append(
                            "<i class='fas fa-edit editTut' data-docid="+doctor.docID+" '></i>"
                            "<i class='fas fa-trash deleteTut' data-docid="+doctor.docID+" '></i>"
                        `));
                });
                $("#updateForm").hide();
                $("#submitDoctorBtn").show();
                loadButtons();
                getRegisteredHospitaIDs();
            }
        });
    }
});
}

```

- DC (AJAX) for Save
 - RESTful URL (to the service)
 - Correct method/type
 - Correct data


```
// ----- Insert New Doctor API -----
function createDoctor(data) {
  console.log(data);
  $.ajax({
    url: 'doctorAPI',
    method: 'POST',
    dataType: 'json',
    data: data,
    success: function(state) {
      console.log(state);

      var stateGot = state;

      if(stateGot > 0){
        getRegisteredDoctors();
        $.notify("Data Successfully Submitted.", "success");
      }
      if(stateGot == 0){
        $.notify("Doctor ID already Extist.", "error");
      }
    },
  });
}
//-----
```

- Alerts

```
$.notify("Data Successfully Submitted.", "success");
```

Or

```
$.notify("Doctor ID already Extist.", "error");
```

- Update button click event handler (event, selector, function)

```
// ----- Update Button Click Event -----
$("#updateDoctorBtn").on("click", function(e) {

    let data = {
        docID: $("#newForm")[0].docID.val(),
        docFname: $("#newForm")[0].docFname.val(),
        docLname: $("#newForm")[0].docLname.val(),
        docPosition: $("#newForm")[0].docPosition.val(),
        docFee: $("#newForm")[0].docFee.val(),
        docAddress: $("#newForm")[0].docAddress.val(),
        mobileNo: $("#newForm")[0].mobileNo.val(),
        hosID: parseInt($("#newForm")[0].hosID.val())+1
    }

    // Form validation-----
    var status = validateItemForm();
    // If not valid-----
    if (status != true)
    {
        $.notify(status, "error");
        $("#updateDoctorBtn").show();
    }

    if(status == true){
        updateDoctorDetails($("#newForm")[0].docID.val(), data);
    }

    $("#newForm").trigger("reset");
    $("#updateDoctorBtn").hide();
    $("#submitDoctorBtn").show();
    e.preventDefault();

    });
});
// -----
```

- Form validation Alert

```
$.notify(status, "error");
```

- Update button click event handling process
 - Read ID
 - Read data
 - Assign to the form

```
// ----- Buttons Initialization-----
function loadButtons() {
    $(".editTut").click(function(e){
        getSingleDoctor($(".this")[0]).data("docid"));
        $("#submitDoctorBtn").hide();
        $("#updateDoctorBtn").show();

        $("#docID").prop("readonly",true);
        e.preventDefault();
    });
}
```

```
//---- Get Selected Doctor details and fill Update fields -----
function getSingleDoctor(id){

    console.log(id);
    var data = {id};

    $.ajax({
        url: 'webapi/doctors/doctor/' + id,
        method: 'get',
        dataType: 'json',
        success: function(data) {
            console.log(data);
            $(".newForm")[0].docID.val(data.docID);
            $(".newForm")[0].docFname.val(data.docFname);
            $(".newForm")[0].docLname.val(data.docLname);
            $(".newForm")[0].docPosition.val(data.docPosition);
            $(".newForm")[0].docFee.val(data.docFee);
            $(".newForm")[0].docAddress.val(data.docAddress);
            $(".newForm")[0].mobileNo.val(data.mobileNo);

            var listItems= "";
            listItems+= "<option value=" + data.hosID + ">" + data.hosID + "</option>";
            $("#hosID").html(listItems);
            $("#updateForm").show();
            $("#editHostId").show();
        }
    });
}

//-----
```

```
//----- Update doctor API -----
function updateDoctorDetails(id, data){
  console.log(data);
  $.ajax({
    url: 'doctorAPI',
    method: 'PUT',
    dataType: 'json',
    data: data,
    success: function(state)
    {
      $.notify("Data Successfully Updated.", "success");
      getRegisteredDoctors();
      $("#docID").prop("readonly", false);
      $("#editHostId").hide();
    }
  });
}
//-----
```

- Alerts

```
$.notify("Data Successfully Updated.", "success");
```

- Remove/Delete button click handler (event, selector, function)

```
$(".deleteTut").click(function(e){
  deleteDoctorDetails($("#this")[0].data("docid"));
  e.preventDefault();
})
//-----
```

- DC (AJAX) for Remove/Delete
 - RESTful URL (to the service)
 - Correct method/type
 - Correct data

```

//----- Delete Doctor details API -----
function deleteDoctorDetails(id){
    console.log(id);
    var data = {id};
    $.ajax({
        url: 'doctorAPI',
        method: 'DELETE',
        dataType: 'json',
        data: data,
        success: function(data) {
            console.log(data);
            getRegisteredDoctors();
            $.notify("Doctor Details Delete Completed.", "warn");

            $("#updateDoctorBtn").hide();
            $("#submitDoctorBtn").show();
        }
    });
}
//-----

```

- Alerts

```
$.notify("Doctor Details Delete Completed.", "warn");
```

Client-Model

- Form validation function
 - Validate elements
 - Return proper error message
 - Return true if valid
 - Completeness

```

/**
 * Servlet implementation class DoctorAPIController
 */
@WebServlet("/doctorAPI")
public class DoctorAPIController extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public DoctorAPIController() {
        super();
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException
    {
        //Getting Registered Hospitals
        DoctorService d = new DoctorService();
        Gson gson = new Gson();
        List<Object> hostpitals = new ArrayList<Object>();
        for( Integer i : d.getAllHospitals()) {
            hostpitals.add(i);
        }
        PrintWriter out = response.getWriter();
        out.print(gson.toJson(hostpitals));
        out.flush();
        out.close();
    }
}

```

```

protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

    DoctorService d = new DoctorService();

    List<Object> doc = new ArrayList<>();

    Map<String, String> docDetails = getParasMap(request);

    doc.add(Integer.parseInt(docDetails.get("docID").toString()));
    doc.add(docDetails.get("docFname").toString());
    doc.add(docDetails.get("docLname").toString());
    doc.add(docDetails.get("docPosition").toString());
    doc.add(Double.parseDouble(docDetails.get("docFee").toString()));
    doc.add(Integer.parseInt(docDetails.get("mobileNo").toString()));
    doc.add(docDetails.get("docAddress").toString());
    doc.add(Integer.parseInt(docDetails.get("hosID").toString()));

    if(d.Create(doc))
    {
        response.getWriter().write("1");
    }else {
        response.getWriter().write("0");
    }
}

```

```

protected void doPut(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

    DoctorService d = new DoctorService();
    List<Object> doc = new ArrayList<>();

    Map<String, String> docUpdateDetails = getParasMap(request);

    doc.add(Integer.parseInt(docUpdateDetails.get("docID").toString()));
    doc.add(docUpdateDetails.get("docFname").toString());
    doc.add(docUpdateDetails.get("docLname").toString());
    doc.add(docUpdateDetails.get("docPosition").toString());
    doc.add(Double.parseDouble((docUpdateDetails.get("docFee").toString())));
    doc.add(Integer.parseInt(docUpdateDetails.get("mobileNo").toString()));
    doc.add(docUpdateDetails.get("docAddress").toString());
    doc.add(Integer.parseInt(docUpdateDetails.get("hosID").toString()));

    System.out.println(doc.toString());
    d.Update(doc);
    response.getWriter().write("true");

}

```

```

protected void doDelete(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException
{

    DoctorService repo = new DoctorService();
    Map<String, String> docDetails = getParasMap(request);
    int deleteID = Integer.parseInt(docDetails.get("id").toString());

    System.out.println("I'm In Delete API: " + deleteID );
    repo.kill(deleteID);
    response.getWriter().write("true");

}

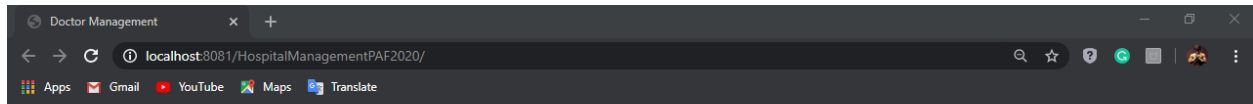
```

```

private static Map<String, String> getParasMap(HttpServletRequest request) {
    Map<String, String> map = new HashMap<String, String>();
    try {
        Scanner scanner = new Scanner(request.getInputStream(), "UTF-8");
        String queryString = scanner.hasNext() ? scanner.useDelimiter("\\A").next() : "";
        scanner.close();
        String[] params = queryString.split("&");
        for (String param : params) {

            String[] p = param.split("=");
            map.put(p[0], p[1]);
        }
    } catch (Exception e) {
    }
    return map;
}

```



Doctor Management

Doctor ID

Firstname

Lastname

Position

Fee

Address

Mobile

Hospital ID

[SubmitDetails](#)

Doc ID	Doctor Firstname	Doctor Lastname	Doctor Position	Doctor Fee	Doctor Address	Doctor Mobile	Doctor Hospital ID	
1	Henry	Thomas	Eye Specialist	1200	11/C15, Kadawatha	779098399	1	Edit Delete
2	Ishan	Alvis	Dentist	2500	1/E2, Kelaniya	711098392	1	Edit Delete
3	Sahan	Medagedara	Neurosurgeon	2500	143/C, Rathnepura	769898399	2	Edit Delete
4	Pasindu	Bandula	Brain Surgeon	2000	133/A, Colombo	726998399	3	Edit Delete



Doctor Management

Doctor ID

Firstname

Lastname

Position

Fee

Address

Mobile

Hospital ID [Edit](#)

[UpdateDetails](#)

Doc ID	Doctor Firstname	Doctor Lastname	Doctor Position	Doctor Fee	Doctor Address	Doctor Mobile	Doctor Hospital ID	
1	Henry	Thomas	Eye Specialist	1200	11/C15, Kadawatha	779098399	1	Edit Delete
2	Ishan	Alvis	Dentist	2500	1/E2, Kelaniya	711098392	1	Edit Delete
3	Sahan	Medagedara	Neurosurgeon	2500	143/C, Rathnepura	769898399	2	Edit Delete
4	Pasindu	Bandula	Brain Surgeon	2000	133/A, Colombo	726998399	3	Edit Delete