|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **□ 수행평가 - 빅데이터를 활용한 IoT 시스템 개발** | | | | | | |
|  |  |  | |  | |  |
| **과정명** | | 빅데이터를 활용한 IoT시스템 개발 | | | | |
| **교과목명** | | 운영관리시스템구축기술 | | **훈련교사** | | 이진만 |
| **과정명** | | Dynamic Web Client 기술이해  Web Server 기술 이해 | | | | |
| **수행날짜** | | 2020.8.21 | 훈련생명 | |  | |
| **과제개요** | | | | | | |
| * Web Client 기술인 HTML5, CSS3.0, JavaScript 와 Servlet 기술을 이용한 시스템 구축 기획 * Web Client 기술을 이용한 다양한 이벤트 처리 및 화면 변환 처리 구현 * 서버와의 연동을 통한 데이터 전송 및 수신 * Open API 연동을 통한 데이터 수신 및 화면 출력 * AJAX를 이용한 데이터 비동기 통신 | | | | | | |
| **1.기획내용**  - POS SYSTEM을 통한 결제 및 관리자 시스템을 구축한다.  - 메뉴판에서 햄버거와 음료수,지점 안내를 클릭할 시 선택한 품목 또는 지점 검색 화면이 뜬다.  - 햄버거와 음료수를 선택하면, 햄버거와 음료수를 선택할 수 있는 화면이 뜨며 선택할 경우 장바구니에 담긴다.  - 2번과 동시에 총액이 자동으로 계산되며,주문하기 버튼을 통해 주문완료.  - 지점 안내 버튼을 누르면, 검색화면과 지도 화면이 구현된다.  - 검색화면에 "맥도날드"를 검색하면 근처 맥도날도 지점이 검색이 된다.  - 검색이 된 지점을 누를 경우 위치와 경도를 확인 후 지도에 마크와 함께 이동한다.  - 관리자 모드 버튼을 클릭 시 로그인 화면으로 바뀌고, 아이디 비밀번호를 입력하면 관리자 화면이 구현된다.  - 관리자 모드에서는 '금일 판매 실적'과 '관리자 설정' 기능을 구현한다.  - 관리자 설정에서는 관리자 등록, 수정, 삭제, 조회 기능을 구현한다.  **2.시스템 구성도**    **3.구현기술.(Java- 8버전, Oracle - 11g,Apache-Tomcat)**  - HTML5, CSS, JavaScript를 활용해 웹 페이지 화면 구축  - Web Application Server 구축을 위해 Apache-Tomcat을 활용  - Servlet, JSP를 활용한 서버연동과 이를 통한 데이터 전송 및 수신  - Oracle Database를 활용한 데이터베이스 구축 및 서버와의 연동  - Kakao Local API를 활용한 Map화면 구성  - AJAX를 이용한 데이터 비동기 통신 구성  **4.구현화면**  [초기화면]      [햄버거 주문, 음료수 주문 화면]      [선택 후 카트에 담기는 화면]    [지점 안내 초기화면]    [지점찾기 후 지점 지도 안내]      [관리자 초기 화면]    [관라지 ID 및 PWD 로그인 화면]    [금일 판매 실적]    [관리자 조회 화면]    [관리자 등록 화면]      [관리자 삭제 화면]      [관리자 정보 수정 화면]      <코드 기능>  [시작]  $(document).ready(function(){    $('.category>.act1').click(function(){  $('.menu').empty();  location.href='chart.hbg';  });  function display(datas){  $(datas).each(function(index,data){  var result = '';  result += '<div num="'+index+'">'  result += '<p>'+data.name+'</p>';  result += '<p>'+data.price+'</p>';  result += '<img src="pic/'+data.img+'">'  result += '</div>'  $('.menu').append(result);  });  [장바구니에 담기기]  $('.menu > div').click(function(){  var num = $(this).attr('num');  cart(num);  });  };    function cart(n){  var name = products[n].name;  var price = products[n].price;  var img = products[n].img;  var result = '';  result += '<img src = "pic/'+img+'">';  result += name + ' ' + price + '<br>';  $('.cart').append(result);  sum += price;  $('.sum > span').text(sum)    };  [햄버거 DB 불러오기]  function getData1(){  $.ajax({  url:'hamburg.hbg',  success: function(result){  products = result;  display(result);  },  error :function(){  alert("Error ...")  }  });  };  $(document).ready(function(){  $('.category>.act1').click(function(){ // 햄버거 메뉴 출력  $('.menu').empty();  getData1();  });  $('.category>.act2').click(function(){  [음료수 DB 불러오기]  function getData2(){  $.ajax({  url:'drink.hbg',  success: function(result){  products = result;  display(result);  },  error: function(){  alert("Error ...")  },  });  };  $('.menu').empty();  getData2();    });  $('.order').click(function(){  tranj(order);      });  });    [지점 검색 기능]  function display3(data){  var datas = $(data).find('documents');  $(datas).each(function(index,d){  var result ='';  result += '<h2>'+$(d).find('address\_name').text()+'</h2>';  $('#result').append(result);  });  };  [DispatcherServlet]  **package** com.sds;  **import** java.io.IOException;  **import** javax.servlet.RequestDispatcher;  **import** javax.servlet.ServletException;  **import** javax.servlet.annotation.WebServlet;  **import** javax.servlet.http.HttpServlet;  **import** javax.servlet.http.HttpServletRequest;  **import** javax.servlet.http.HttpServletResponse;  @WebServlet({ "/DispatcherServlet", "/dispatcher" })  **public** **class** DispatcherServlet **extends** HttpServlet {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;  **public** DispatcherServlet() {  **super**();  }  **protected** **void** service(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  request.setCharacterEncoding("EUC-KR");  String uri = request.getRequestURI();  String path = uri.substring(uri.lastIndexOf("/"));  path = path.substring(1, path.lastIndexOf("."));  System.***out***.println("request:"+path);  String next = "main.jsp";  **if**(path.equals("main")) {  next = "main.jsp";  }**else** **if**(path.equals("administer")) {  next = "administer.jsp";  }**else** **if**(path.equals("hamburg")) {  next = path;  }**else** **if**(path.equals("drink")) {  next = path;  }**else** **if**(path.equals("map")){  next = "map.jsp";  }**else** **if**(path.equals("admin\_login")) {  next = "admin\_login.jsp";  }**else** **if**(path.equals("chart")) {  next = "administer.jsp";  request.setAttribute("centerpage", "chart");  }**else** **if**(path.equals("weather")) {  next = "administer.jsp";  request.setAttribute("centerpage", "weather");  }**else** **if**(path.equals("register")) {  next = "administer.jsp";  request.setAttribute("centerpage", "register");  }**else** {  next =path;  }  RequestDispatcher rd =  request.getRequestDispatcher(next);  rd.forward(request, response);  }  }  [햄버거 메뉴를 출력하는 Servlet]  package com.sds;  import java.io.IOException;  import java.io.PrintWriter;  import java.util.ArrayList;  import javax.servlet.RequestDispatcher;  import javax.servlet.ServletException;  import javax.servlet.annotation.WebServlet;  import javax.servlet.http.HttpServlet;  import javax.servlet.http.HttpServletRequest;  import javax.servlet.http.HttpServletResponse;  import org.json.simple.JSONArray;  import org.json.simple.JSONObject;  import controller.HamburgController;  import frame.Controller;  import vo.Hamburg;  @WebServlet({ "/HamburgServlet", "/hamburg" })  public class HamburgServlet extends HttpServlet {  private static final long serialVersionUID = 1L;  Controller<String,Hamburg> controller;    public HamburgServlet() {  super();  controller = new HamburgController();  }  protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  ArrayList<Hamburg> list = null;  try {  list = controller.get();  JSONArray ja = new JSONArray();  for(Hamburg u:list) {  JSONObject obj = new JSONObject();  obj.put("name", u.getName());  obj.put("price", u.getPrice());  obj.put("img", u.getImage());    ja.add(obj);  }    response.setContentType("text/json;charset=euc-kr");  PrintWriter out = response.getWriter();  out.print(ja.toJSONString());  } catch (Exception e) {  e.printStackTrace();  }  }  }  [음료수 메뉴를 출력하는 Servlet]  **package** com.sds;  **import** java.io.IOException;  **import** java.io.PrintWriter;  **import** java.util.ArrayList;  **import** javax.servlet.ServletException;  **import** javax.servlet.annotation.WebServlet;  **import** javax.servlet.http.HttpServlet;  **import** javax.servlet.http.HttpServletRequest;  **import** javax.servlet.http.HttpServletResponse;  **import** org.json.simple.JSONArray;  **import** org.json.simple.JSONObject;  **import** controller.DrinkController;  **import** frame.Controller;  **import** vo.Drink;  @WebServlet({ "/DrinkServlet", "/drink" })  **public** **class** DrinkServlet **extends** HttpServlet {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;  Controller<String,Drink> controller;  **public** DrinkServlet() {  **super**();  controller = **new** DrinkController();  }  **protected** **void** service(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  ArrayList<Drink> list = **null**;  **try** {  list = controller.get();  JSONArray ja = **new** JSONArray();  **for**(Drink u:list) {  JSONObject obj = **new** JSONObject();  obj.put("name", u.getName());  obj.put("price", u.getPrice());  obj.put("img", u.getImage());  ja.add(obj);  }  response.setContentType("text/json;charset=euc-kr");  PrintWriter out = response.getWriter();  out.print(ja.toJSONString());  } **catch** (Exception e) {  e.printStackTrace();  }  }  }  [지점 검색 화면]  <script  src=*"https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"*></script>  <script type=*"text/javascript"*  src=*"//dapi.kakao.com/v2/maps/sdk.js?appkey=a3c7e75cddfe51d0cab88acd296b8fdf"*></script>  **var** map = **null**;  **var** place = **null**;  **function** maps(n){  **var** doc = place.documents;  **var** positionx = parseFloat(doc[n].x);  **var** positiony = parseFloat(doc[n].y);  alert(**typeof**(positionx)+' ' +**typeof**(positiony));  **var** moveLatLon = **new** kakao.maps.LatLng(positiony, positionx);  // 지도 중심을 이동 시킵니다  map.setCenter(moveLatLon);  **var** markerPosition = **new** kakao.maps.LatLng(positiony, positionx);  // 마커를 생성합니다  **var** marker = **new** kakao.maps.Marker({  position : markerPosition  });  // 마커가 지도 위에 표시되도록 설정합니다  marker.setMap(map);  }  **function** display(data){  $(data.documents).each(**function** (index,add){  **var** result = '';  result += '<div num="'+index+'">';  result += '<p>'+add.place\_name+'</p>';  result += '</div>';  $('#result').append(result);  });  $('#result > div').click(**function**(){  **var** num = $(**this**).attr('num');  maps(num);  });  };  **function** getData(){    **var** urlstr = 'https://dapi.kakao.com/v2/local/search/keyword.json?y=37.514322572335935&x=127.06283102249932&radius=20000';  **var** loc = $('input[name = "loc"]').val();    $.ajax({  method:'GET',  url : urlstr,  headers : {'Authorization' : 'KakaoAK 8e1649490830f8284eaa7a508cf61f2d'},  type : 'GET',  data:{'query':loc},  asynch:**false**,  success:**function**(data){  //alert(data);  display(data);  place = data;  console.log(place);  },  error:**function**(){  alert('error');  }  });  };  $(document).ready(**function**(){  **var** container = document.getElementById('map\_start');  **var** options = {  center : **new** kakao.maps.LatLng(37.557013, 126.992683),  level : 8  };  map = **new** kakao.maps.Map(container, options);  $('button').click(**function**(){  $('#result').empty();  getData();  });  });  [관리자 페이지 로그인 Servlet]  @WebServlet({ "/LoginImplServlet", "/loginimpl" })  **public** **class** LoginImplServlet **extends** HttpServlet {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;  Controller<String, Users> controller;    **public** LoginImplServlet() {  **super**();  controller = **new** UsersController();  }  **protected** **void** service(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {    String id = request.getParameter("id");  String pwd = request.getParameter("pwd");  // DB Matching  Users dbuser = **null**;  **try** {  dbuser = controller.get(id);  **if**(dbuser.getPwd().equals(pwd)) {  response.sendRedirect("administer.jsp");  }**else** {  request.setAttribute("centerpage", "admin\_login");  }  } **catch** (Exception e) {  request.setAttribute("centerpage", "loginfail");  e.printStackTrace();  }  }  }  [관리자 화면 기능]  **function** getChart(){  $.ajax({  url:'chart.hbg',  success: **function**(result){  **return** result;  },  error :**function**(){  alert("Error ...")  },  });  };  $(document).ready(**function**(){    $('.category>.act1').click(**function**(){  $('.menu').empty();  location.href='chart.hbg';  });  [차트 기능]  <script src=*"https://code.highcharts.com/highcharts.js"*></script>  <script src=*"https://code.highcharts.com/modules/exporting.js"*></script>  <script src=*"https://code.highcharts.com/modules/export-data.js"*></script>  <script src=*"https://code.highcharts.com/modules/accessibility.js"*></script>  <script type=*"text/javascript"*>  Highcharts.chart('container', {  chart: {  plotBackgroundColor: **null**,  plotBorderWidth: **null**,  plotShadow: **false**,  type: 'pie'  },  title: {  text: '맥도날드 판매 내역 및 품목'  },  tooltip: {  pointFormat: '{series.name}: <b>{point.percentage:.1f}%</b>'  },  accessibility: {  point: {  valueSuffix: '%'  }  },  plotOptions: {  pie: {  allowPointSelect: **true**,  cursor: 'pointer',  dataLabels: {  enabled: **true**,  format: '<b>{point.name}</b>: {point.percentage:.1f} %'  }  }  },  series: [{  name: 'Brands',  colorByPoint: **true**,  data: [{  name: '새우버거',  y: 9.5,  sliced: **true**,  selected: **true**  }, {  name: '불고기버거',  y: 4.3  }, {  name: '상하이버거',  y: 7.6  }, {  name: '슈슈버거',  y: 14.1  }, {  name: '슈비버거',  y: 24.5  }, {  name: '코카콜라',  y: 19.8  }, {  name: '코카제로',  y: 2.2  }, {  name: '스프라이트',  y: 8.9    },{  name: '펩시',  y: 3.3  }, {  name: '환타',  y: 5.8  }]  }]  });  [관리자 명단 출력 Servlet]  package com.sds;  import java.io.IOException;  import java.util.ArrayList;  import javax.servlet.RequestDispatcher;  import javax.servlet.ServletException;  import javax.servlet.annotation.WebServlet;  import javax.servlet.http.HttpServlet;  import javax.servlet.http.HttpServletRequest;  import javax.servlet.http.HttpServletResponse;  import controller.UsersController;  import frame.Controller;  import vo.Users;  @WebServlet({ "/SelectAllServlet", "/selectall" })  public class SelectAllServlet extends HttpServlet {  private static final long serialVersionUID = 1L;    Controller<String, Users> controller;    public SelectAllServlet() {  super();  controller = new UsersController();  }  protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  // 1. Database에서 데이터를 가지고 온다.  ArrayList<Users> list = null;  try {  list = controller.get();  } catch (Exception e) {  e.printStackTrace();  }  // 2. 가지고 온 데이터를 request 에 셋팅 한다.  // 3. main 페이지로 이동한다.  // 4. centerpage 를 셋팅한다.  request.setAttribute("userlist", list);  request.setAttribute("centerpage", "selectall");  RequestDispatcher rd =  request.getRequestDispatcher("administer.jsp");  rd.forward(request, response);  }  }  [관리자 상세 Servlet]  @WebServlet({ "/UserDetailServlet", "/userdetail" })  **public** **class** UserDetailServlet **extends** HttpServlet {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;  Controller<String, Users> controller;    **public** UserDetailServlet() {  **super**();  controller = **new** UsersController();  }  **protected** **void** service(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  String id = request.getParameter("id");  Users dbuser = **null**;  **try** {  dbuser = controller.get(id);  request.setAttribute("userdetail", dbuser);  request.setAttribute("centerpage", "userdetail");  } **catch** (Exception e) {  e.printStackTrace();  }  RequestDispatcher rd =  request.getRequestDispatcher("administer.jsp");  rd.forward(request, response);  }  }  [관리자 삭제 Servlet]  @WebServlet({ "/UserDeleteServlet", "/userdelete" })  **public** **class** UserDeleteServlet **extends** HttpServlet {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;  Controller<String, Users> controller;  **public** UserDeleteServlet() {  **super**();  controller = **new** UsersController();  }  **protected** **void** service(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  String id = request.getParameter("id");  **try** {  controller.remove(id);  response.sendRedirect("selectall");  } **catch** (Exception e) {  e.printStackTrace();  }  }  }  [관리자 정보 수정 페이지 이동 Servlet]  @WebServlet({ "/UserUpdateServlet", "/userupdate" })  **public** **class** UserUpdateServlet **extends** HttpServlet {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;  Controller<String, Users> controller;  **public** UserUpdateServlet() {  **super**();  controller = **new** UsersController();  }  **protected** **void** service(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  String id = request.getParameter("id");  Users dbuser = **null**;  **try** {  dbuser = controller.get(id);  request.setAttribute("dbuser", dbuser);  request.setAttribute("centerpage", "userupdate");  } **catch** (Exception e) {  e.printStackTrace();  }  RequestDispatcher rd =  request.getRequestDispatcher("administer.jsp");  rd.forward(request, response);  }  }  [관리자 정보 수정 Servlet]  @WebServlet({ "/UserUpdateImplServlet", "/userupdateimpl" })  **public** **class** UserUpdateImplServlet **extends** HttpServlet {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;  Controller<String, Users> controller;  **public** UserUpdateImplServlet() {  **super**();  controller = **new** UsersController();  }  **protected** **void** service(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  String id = request.getParameter("id");  String pwd = request.getParameter("pwd");  String name = request.getParameter("name");  String job = request.getParameter("job");    Users user = **new** Users(id, pwd, name, job);    **try** {  controller.modify(user);  response.sendRedirect("userdetail.hbg");  } **catch** (Exception e) {  e.printStackTrace();  }  }  }  [관리자 등록 Servlet]  @WebServlet({ "/RegisterImplServlet", "/registerimpl" })  **public** **class** RegisterImplServlet **extends** HttpServlet {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;  Controller<String, Users> controller;    **public** RegisterImplServlet() {  **super**();  controller = **new** UsersController();  }  **protected** **void** service(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {    String id = request.getParameter("id");  String pwd = request.getParameter("pwd");  String name = request.getParameter("name");  String job = request.getParameter("job");    Users users = **new** Users(id, pwd, name, job);    **try** {  controller.register(users);  response.sendRedirect("selectall.hbg");  } **catch** (Exception e) {  request.setAttribute("centerpage", "registerfail");  e.printStackTrace();  }  }  } | | | | | | |