Account controller

using ClassLibrary.WebModels;

using EntitiesLibrary;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace AirbncleanWeb.Controllers

{

public class AccountController : PreLoginBaseController

{

/// <summary>

/// register

/// </summary>

/// <returns></returns>

public ActionResult register()

{

return View();

}

[HttpPost]

public object register(RegisterModel model)

{

bool status = false;

string message = "Failed";

try

{

Tuple<bool, string, tRegistration, ClassLibrary.Enum.UserRole> t = \_repository.register(model);

status = t.Item1;

message = t.Item2;

if (status)

{

Session["AccountDetails"] = t.Item3;

return Json(new { Status = status, Message = message, Type = (int)t.Item4 }, JsonRequestBehavior.AllowGet);

}

else

return Json(new { Status = status, Message = message }, JsonRequestBehavior.AllowGet);

}

catch (Exception ex)

{

return new { Status = status, Message = ex.Message };

}

}

public ActionResult login(int type)

{

var model = new LogInModel();

model.type = type;

return View(model);

}

[HttpPost]

public object login(LogInModel model)

{

bool status = false;

string message = "Failed";

try

{

Tuple<bool, string, tRegistration> t = \_repository.login(model);

status = t.Item1;

message = t.Item2;

if (status)

{

Session["AccountDetails"] = t.Item3;

return Json(new { Status = status, Message = message, Type = model.type }, JsonRequestBehavior.AllowGet);

}

else

return Json(new { Status = status, Message = message }, JsonRequestBehavior.AllowGet);

}

catch (Exception ex)

{

return new { Status = status, Message = ex.Message };

}

}

public ActionResult home()

{

return View();

}

[HttpGet]

public void Refreshtimezone(string id)

{

EntitiesLibrary.Data.BaseReference obj = new EntitiesLibrary.Data.BaseReference(id);

}

[HttpGet]

public object forgotPassword(string email)

{

bool status = false;

string message = "Email not registered";

try

{

Tuple<bool, string> t = \_repository.forgotPassword(email);

status = t.Item1;

message = t.Item2;

if (status)

{

return Json(new { Status = status, Message = message }, JsonRequestBehavior.AllowGet);

}

else

return Json(new { Status = status, Message = message }, JsonRequestBehavior.AllowGet);

}

catch (Exception ex)

{

return new { status = status, message = ex.Message };

}

}

}

}