Asp.Net Project

Description:

ScheduleIt2.0", an ASP.NET  MVC user management system that’s managed under the SCRUM methodology and utilizes Azure Dev Ops. The system will track user clock-ins, schedules, and requests for time off, etc.

My working Code and images after editing the code:

Add TotalHours wich count totalhours of work

/// <summary>

/// result of total hours in shift time (shift end - shift start)

/// </summary>

[Display(Name = "Total Hours")]

[DataType(DataType.Date)]

[DisplayFormat(DataFormatString = @"{0:hh\:mm\:ss}", ApplyFormatInEditMode = true),

Range(typeof(TimeSpan), "00:00", "23:59")]

public TimeSpan? TotalHours

{

get

{

if (StartTime.HasValue && EndTime.HasValue)

{

return EndTime.Value.Subtract(StartTime.Value);

}

return null;

}

set { }

}

Show clocked in status for user

// Grabs the current user ID

var userId = User.Identity.GetUserId();

// Grabs all events in Db that have the same user ID as the one logging in

ApplicationUser user = db.Users.Find(userId);

//Checking for clocked in status

bool ClockedInStatus = db.WorkTimeEventModels.Any(x => x.User.Id == userId && !x.EndTime.HasValue);

if (ClockedInStatus == true)

{

ViewBag.Status = "Currently Clocked In";

}

else

{

ViewBag.Status = "Not Clocked In";

}

var result = (from t in db.WorkTimeEventModels

where t.EventId == t.EventId

select new

{

t.TotalHours

}).ToList();

double sum = 0;

foreach (var item in result)

{

//if (item.TotalHours.HasValue)

//{

// TimeSpan timespan = item.TotalHours.Value;

// sum += item.TotalHours.HasValue ? item.TotalHours.Value.TotalMinutes : 0;

//}

//else

//{

// return null;

//}

//sum += Convert.ToDouble(item.TotalHours);

// sum += item.TotalHours.HasValue ? item.TotalHours.Value.TotalMinutes : 0;

ViewBag.ShowSum = sum;

var projectedpay = Convert.ToDecimal(sum / 60) \* user.HourlyRate;

ViewBag.Pay = projectedpay;

}

**Show total hours and payrate and status**

@using ScheduleIt2.\_0.Models

@model IEnumerable<ScheduleIt2.\_0.Models.WorkTimeEventModel>

@{

ViewBag.Title = "Work Periods";

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<h2>Work Periods</h2>

<div align="right">

Total Hours:

@Html.ViewBag.ShowSum

<br />

Status:

@Html.ViewBag.Status

<br />

Projected Pay:

@Html.ViewBag.Pay

</div>

<table class="table">

<tr>

<th>

@Html.ActionLink("Shift Start Date", "Index", new { sortOrder = ViewBag.StartDateSortParm, currentFilter = ViewBag.CurrentFilter })

</th>

<th>

@Html.ActionLink("Shift End Date", "Index", new { sortOrder = ViewBag.EndDateSortParm, currentFilter = ViewBag.CurrentFilter })

</th>

<th>

@Html.ActionLink("Total Hours", "Index", new { sortOrder = ViewBag.TotalHoursSortParmd, currentFilter = ViewBag.CurrentFilter })

</th>

<th>

Shift Notes

</th>

<th></th>

</tr>

@foreach (var item in Model)

{

<tr>

<td>

@Html.DisplayFor(modelItem => item.StartTime)

</td>

<td>

@Html.DisplayFor(modelItem => item.EndTime)

</td>

<td>

@Html.DisplayFor(modelItem => item.TotalHours)

</td>

<td>

@Html.DisplayFor(modelItem =>item.Message)

</td>

<td>

@Html.ActionLink("Edit", "Edit", new { id = item.EventId })

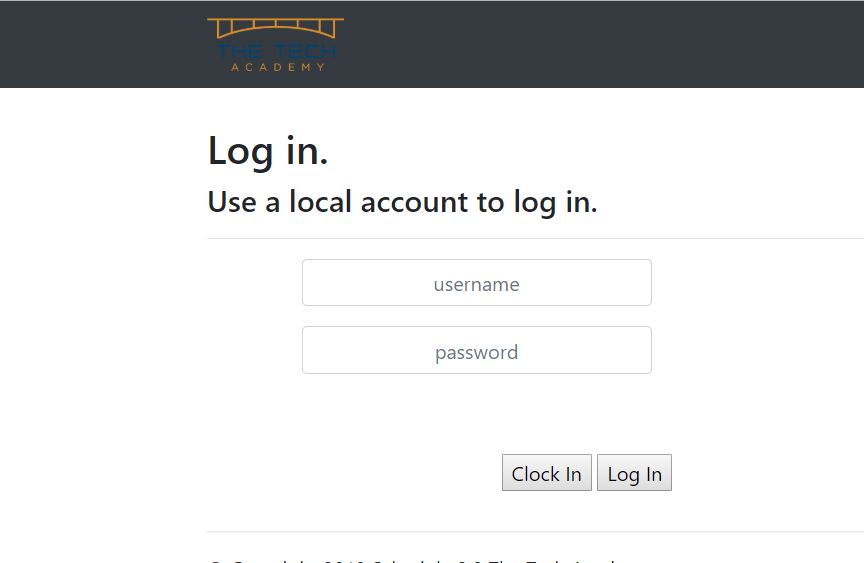
</td>

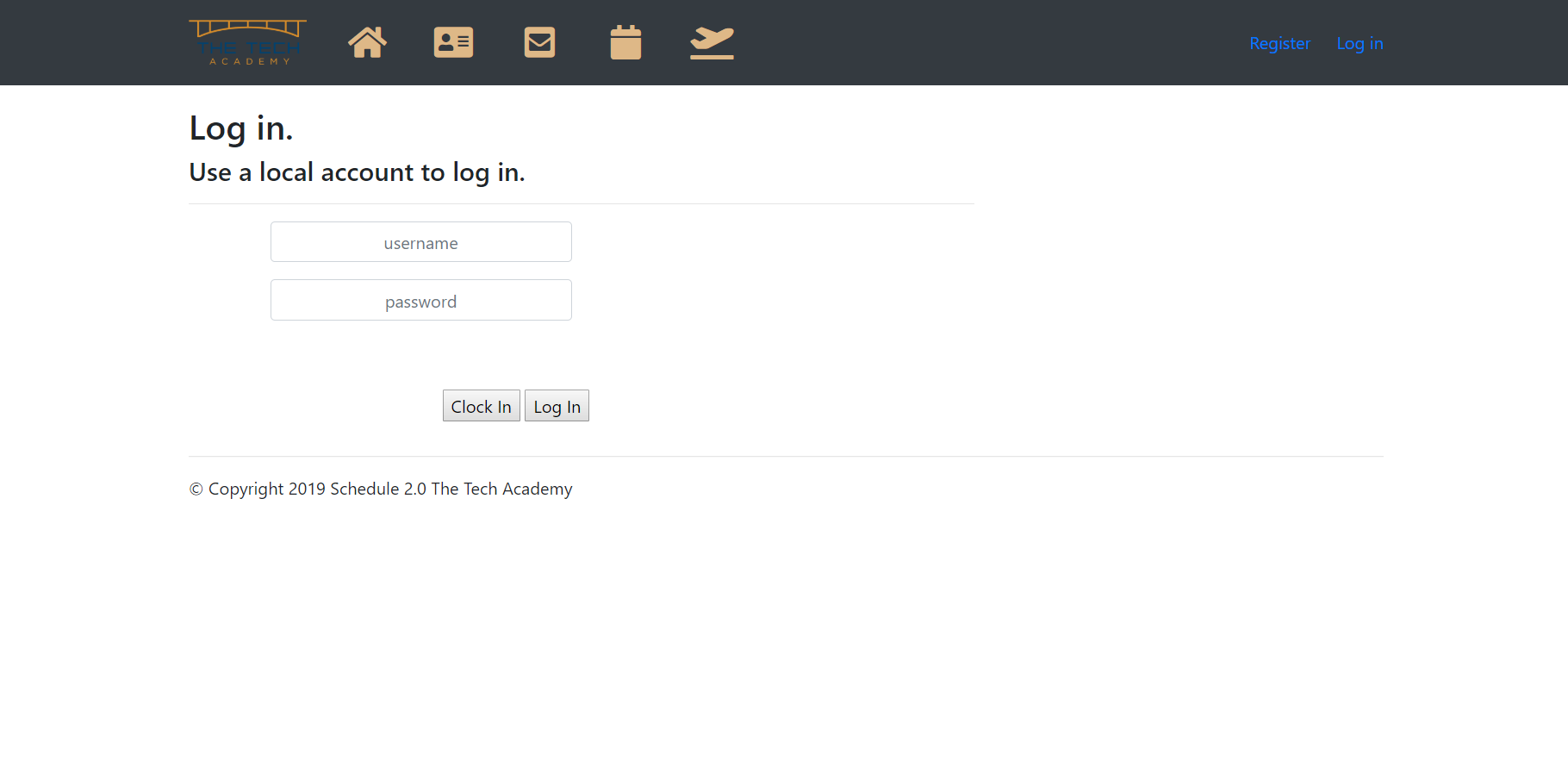
</tr>

}

</table>

Change login page





<ul class="nav navbar-nav">

<li>

<a href="@Url.Action("Index", "Home")">

<span class="fa-stack fa-2x">

<i class="fas fa-home fa-stack-1x fa-inverse" style="color:burlywood"></i>

</span>

@\*Home\*@

</a>

</li> <!--class="nav-item active mr-3" added too give links space-->

<li>

<a href="@Url.Action("About", "Home")">

<span class="fa-stack fa-2x">

<i class="fas fa-address-card fa-stack-1x fa-inverse" style="color:burlywood"></i>

</span>

@\*About\*@

</a>

</li>

<li>

<a href="@Url.Action("Contact", "Home")">

<span class="fa-stack fa-2x">

<i class="fas fa-envelope-square fa-stack-1x fa-inverse" style="color:burlywood"></i>

</span>

@\*Contact\*@

</a>

</li>

<li>

<a href="@Url.Action("Index", "Calendar")">

<span class="fa-stack fa-2x">

<i class="fa fa-calendar fa-stack-1x fa-inverse" style="color:burlywood"></i>

</span>

@\*Calendar\*@

</a>

</li>

<li>

<a href="@Url.Action("Index", "TimeOffEvent")">

<span class="fa-stack fa-2x">

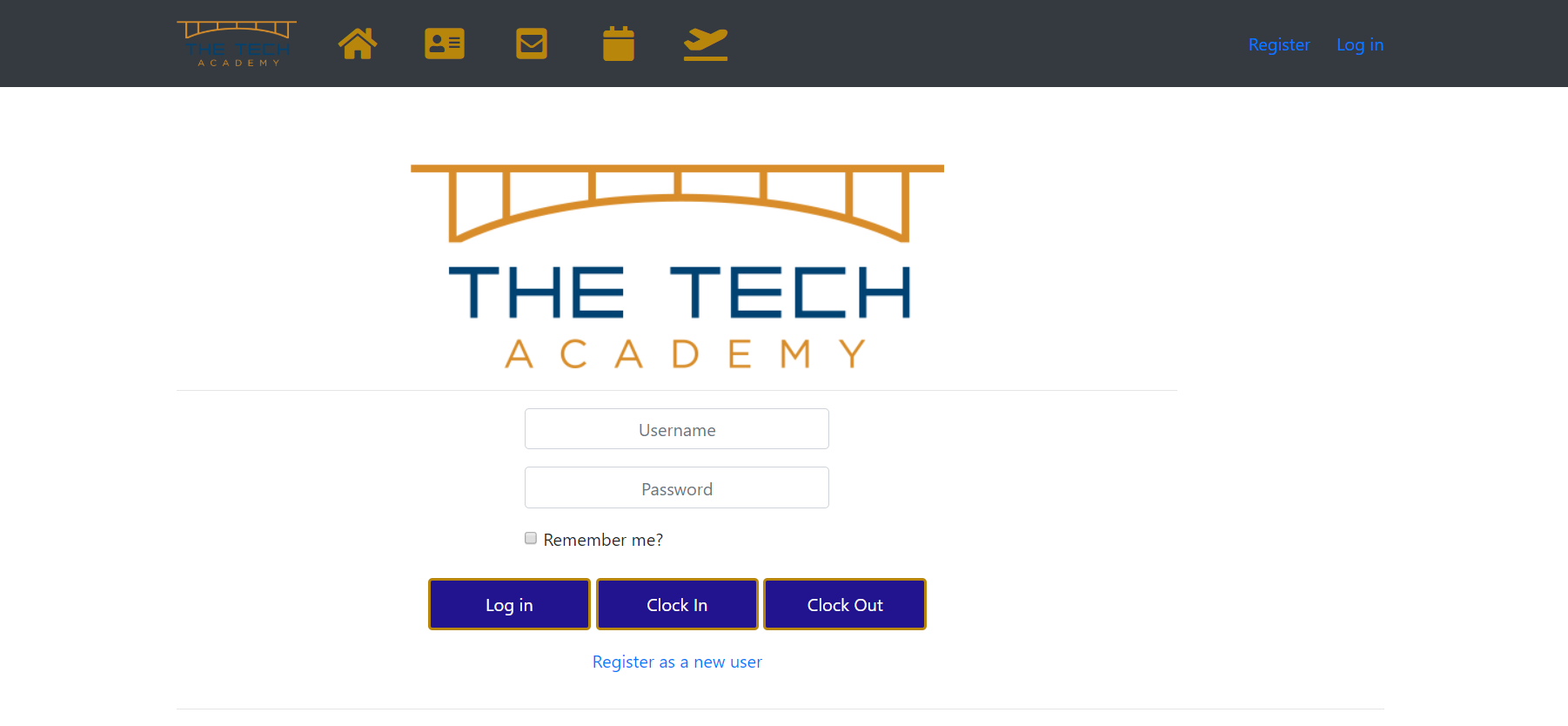
<i class="fa fa-plane-departure fa-stack-1x fa-inverse" style="color:burlywood"></i>

</span>

</a>

</li>

**Change login**



body {

padding-top: 50px;

padding-bottom: 50px;

}

/\* Set padding to keep content from hitting the edges \*/

.body-content {

padding-left: 15px;

padding-right: 15px;

}

/\* Override the default bootstrap behavior where horizontal description lists

will truncate terms that are too long to fit in the left column

\*/

.dl-horizontal dt {

white-space: normal;

}

/\* Set width on the form input elements since they're 100% wide by default \*/

input,

select,

textarea {

max-width: 280px;

}

#logo {

margin: 0 auto;

display: block;

}

#workevent-logo {

float: left;

margin: 2% auto;

}

.my-1 {

margin-top: 1em;

margin-bottom: 1em;

}

.centerthis {

margin-left: auto;

margin-right: auto;

}

.clockinbtn {

background-color: #22148e;

border: solid;

border-color: darkgoldenrod;

border-width: 2px;

border-radius: 4px;

color: white;

padding: 10px 23px;

text-align: center;

text-decoration: none;

display: inline-block;

font-size: 16px;

cursor: pointer;

width: 150px;

}

.clockinbtn:hover {

background-color: darkgoldenrod;

border-color: #1A2B51;

color: white;

}

.loginbtn {

background-color: #22148e;

border: solid;

border-color: darkgoldenrod;

border-width: 2px;

border-radius: 4px;

color: white;

padding: 10px 23px;

text-align: center;

text-decoration: none;

display: inline-block;

font-size: 16px;

cursor: pointer;

width: 150px;

}

.loginbtn:hover {

background-color: darkgoldenrod;

border-color: #1A2B51;

color: white;

}

.uPassword {

border-color: blue;

font-weight: bold;

color: #808080;

cursor: pointer;

}

.uPassword:hover {

border-color: #FFB035;

}

.username {

border-color: blue;

font-weight: bold;

color: #808080;

cursor: pointer;

}

.username:hover {

border-color: #FFB035;

}