TypeScript

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

app.ts

/// <reference path="greetingpage.ts" />

class Program {

static Main(): void {

GreetingPage.Start();

VisiblePagesController.HideAll();

ClicksByBtnsController.ObserveAllButtons();

}

}

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

EntitiesDB.ts

class Category {

Id: number = 0;

Name: string;

}

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

GreetingPage.ts

/// <reference path="../@types/jquery/index.d.ts" />

class GreetingPage {

static ClickMoreBtn() {

return $("#getmore").click(() => { GreetingPage.GetMore() });

}

static GetMore() {

$("#getmore").text("Loading...");

let link: string = this.GetLink();

this.GetRequest(link, "GET");

}

static GetCategory() {

return $(".category span strong").text();

}

static GetName() {

return $(".name span strong").text();

}

static GetLink(): string {

return "/api/Events/" + this.GetCategory() + "/" + this.GetName()+"/" + $("#siteEvents .row").length + "/";

}

static GetRequest(link: string, typeRequest: string) {

$.ajax({

url: link,

type: typeRequest,

success: function (data) {

this.GetSuccess(data);

}.bind(this)

});

}

static GetSuccess(data: EventMsg) {

this.ShowListOfMessage(data.List);

this.ChooseMore(data.Msg);

}

static ShowListOfMessage(data: TaskVM[]) {

for (var item of data) {

var date = item.Date;

var str = '<div class = "row">' +

'<div class="col-md-2 col-md-offset-1 s2-column1">' +

'<h2>' + item.TaskName + '</h2>' +

'<span>' + item.Name + '</span>' +

'<span>' + date.toLocaleString() + '</span></div>' +

'<div class = "col-md-8 s2-column2">' +

'<p>' + item.Description + '</p></div></div>';

$("#siteEvents").append(str);

}

}

static ChooseMore(msg: string) {

$("#getmore").text(msg);

var checkMore = $("#getmore").text().toString();

if (checkMore == "There are no more events") {

$("#getmore").off("click");

$("#getmore").css("cursor", "text");

}

}

static ClickByCategoryItem() {

$(".categories li a").click(function (event) {

$(".categories li a").removeClass("selected");

$(event.target).addClass("selected");

var category = $(event.target).text();

$(".category span strong").text(category);

if ($("#siteEvents .row").length > 0) {

$("#siteEvents").text("");

}

$("#getmore").off("click");

$("#getmore").on("click", () => { GreetingPage.GetMore() });

$("#getmore").css("cursor", "pointer");

GreetingPage.GetMore();

});

}

static ClickByNameItem() {

$(".names li a").click(function (event) {

$(".names li a").removeClass("selected");

$(event.target).addClass("selected");

var name = $(event.target).text();

$(".name span strong").text(name);

if ($("#siteEvents .row").length > 0) {

$("#siteEvents").text("");

}

$("#getmore").off("click");

$("#getmore").on("click", () => { GreetingPage.GetMore() });

$("#getmore").css("cursor", "pointer");

GreetingPage.GetMore();

});

}

static ClickByBtnInformation() {

$(".btn-information").click(function () {

if ($(".s1").is(":hidden")) {

$(".s1").show("slow");

$(".btn-information").text("hide");

if (window.matchMedia('(max-width: 480px)').matches) {

$(".s2").css("top", "160px");

$("footer").css("margin-top", "160px")

}

else if (window.matchMedia('(max-width: 768px)').matches) {

$(".s2").css("top", "160px");

$("footer").css("margin-top", "160px")

}

}

else {

$(".s1").hide("slow");

$(".btn-information").text("show");

if (window.matchMedia('(max-width: 480px)').matches) {

$(".s2").css("top", "50px");

$("footer").css("margin-top", "50px")

}

else if (window.matchMedia('(max-width: 768px)').matches) {

$(".s2").css("top", "50px");

$("footer").css("margin-top", "50px")

}

}

});

}

static ClickByHideShowBtnInformation() {

if ($(".s1").is(":hidden")) {

if (window.matchMedia('(min-width: 768px)').matches) {

$("footer").css("margin-top", "0px")

}

else if (window.matchMedia('(max-width: 768px)').matches) {

$(".s2").css("top", "50px");

$("footer").css("margin-top", "50px")

}

else if (window.matchMedia('(max-width: 480px)').matches) {

$(".s2").css("top", "50px");

$("footer").css("margin-top", "50px")

}

}

else {

if (window.matchMedia('(min-width: 768px)').matches) {

$("footer").css("margin-top", "0px")

}

else if (window.matchMedia('(max-width: 768px)').matches) {

$(".s2").css("top", "160px");

$("footer").css("margin-top", "160px")

}

else if (window.matchMedia('(max-width: 480px)').matches) {

$(".s2").css("top", "160px");

$("footer").css("margin-top", "160px")

}

}

}

static ResizeWindow() {

$(window).resize(function () {

GreetingPage.ClickByHideShowBtnInformation();

});

}

static Start() {

$().ready(() => {

this.ClickMoreBtn();

this.ClickByCategoryItem();

this.ClickByNameItem();

this.ClickByBtnInformation();

this.ResizeWindow();

});

}

}

class TaskVM {

public TaskName: string;

public Name: string;

public Description: string;

public Category: string;

public Date: Date;

}

class EventMsg {

public Msg: string = "More";

public List: TaskVM[];

}

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

VisiblePagesController.ts

class VisiblePagesController {

static HideAll() {

$("#CreateCategoryForm").hide();

$("#GreetingPage").hide();

}

static ShowFormCreateCategory() {

VisiblePagesController.HideAll();

if ($("#CreateCategoryForm").is(":hidden")) {

$("#CreateCategoryForm").show();

}

}

static ShowGreetingPage() {

VisiblePagesController.HideAll();

if ($("#GreetingPage").is(":hidden")) {

$("#GreetingPage").show();

}

}

}

class ClicksByBtnsController {

static ObserveAllButtons() {

this.ClickCreateCategoryBtn();

this.ClickShowGreetingPageBtn();

}

static ClickCreateCategoryBtn() {

$("#CreateCategoryBtn").on("click", () => { VisiblePagesController.ShowFormCreateCategory(); })

}

static ClickShowGreetingPageBtn() {

$("#ShowGreetingPageBtn").on("click", () => { VisiblePagesController.ShowGreetingPage(); })

}

}