**Міністерство освіти та науки України**

**Київський національний торговельно-економічний університет Факультет інформаційних технологій**

**Кафедра інженерії програмного забезпечення та кібербезпеки**

**ПРАКТИЧНА РОБОТА  НА ТЕМУ:**

**Інформаційна система MUKILSKO\_INFO**



(назва теми)

Виконав студент факультету

Глижін Владислав Сергійович

Групи 7 курсу 3

Київ 20 22

**Зміст**

[1. Предметна область 3](#_Toc107157121)

[2. Архітектура розроблюваного веб додатку 3](#_Toc107157122)

[2.1. Еталонна модель веб сайту 3](#_Toc107157123)

[2.2. Блок схема веб сайту 4](#_Toc107157124)

[2.3. Use case діаграма 5](#_Toc107157125)

[2.4. Діаграма класів 5](#_Toc107157126)

[2.5. Діаграма MVC архітектури 7](#_Toc107157127)

[2.6. Діаграма стану 9](#_Toc107157128)

[2.7. Діаграма розгортання 10](#_Toc107157129)

[3. Опис бази даних 11](#_Toc107157130)

[3.1. Логічна модель бази даних 11](#_Toc107157131)

[3.2. Фізична модель бази даних 11](#_Toc107157132)

[3.3. Нормалізація бази даних 13](#_Toc107157133)

[3.4. Прив’язка бази даних та веб додатку 14](#_Toc107157134)

[3.5. Скріншоти наповненої бази даних 15](#_Toc107157135)

[4. Дизайн проекту у Figma 16](#_Toc107157136)

[5. Результат роботи (скріншоти) 17](#_Toc107157137)

[6. Додатки 20](#_Toc107157138)

[Додаток А 20](#_Toc107157139)

[Додаток Б 22](#_Toc107157140)

[Додаток В 24](#_Toc107157141)

# **Предметна область**

В Україні починаючи з 2015 року стрімко стали набувати популярності вживані, злегка пошкоджені, однак досить вигідні до придбання, порівнюючи з ринком України, автомобілі з онлайн аукціонів Сполучених Штатів Америки (США) таких як Copart[1], Manheim[2], та IAA-Insurance[3].

Як правило, з США везуть автомобілі, що втрапили в дорожньо-транспортну пригоду (ДТП) або були пошкоджені після повеней чи інших катаклізмів. [5]

Справа в тому, що американці часто беруть автомобілі в лізинг або купують їх за допомогою кредиту. Й в випадках, коли з ними щось трапляється ремонт автомобіля в США може бути занадто коштовним й не вигідним для страхових компаній. Саме в цих випадках транспортні засоби й виставляються на аукціонах де їх й купують іноземці.

Так, авто середнього класу і доброї комплектації можна придбати за $4-5 тис. USD, однак варто зауважити що це лише менша частина витрат на це авто. Аби отримати справний автомобіль на котрому можна пересуватись дорогами України, до вартості авто слід додати: витрати на послуги аукціону, транспортування автомобіля з аукціону до порту, а звідти вже до нашої неньки, далі в черзі йде розмитнення авто, його ремонт й, нарешті, сертифікація (перевірка чи справний автомобіль) й, фінальна, постанова на облік.

Зважаючи на весь цей шлях, котрий подолає бажане авто й час, який на це буде затрачено і ось відповідь на питання «Чому ж українці зважуються на цей шлях ?». Все досить просто – такі авто в сумі коштують дешевше автомобілей, що продаються на українському автомобільному ринку.

Приведемо трохи статистики:

Таблиця 1.1 – Статистика імпортованих авто в Україну

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Рік | 2015 | 2016 | 2017 | 2018 | 2019 | 2020 |
| Кількість автомобілей імпортованих в Україну (шт.) | 62313 | 87768 | 154410 | 230050 | 544043 | 531400 |
| Сума імпортованих автомобілей саме з США (Млн. USD) | - | - | 229,29 | 316,76 | 518,50 | 664,88 |

*Джерело: таблиця побудована автором, дані взяті з офіційного сайту митниці України [4] та джерела [5]*

Ми бачимо що кількість автомобілей, що вперше отримують українські номерні знаки все росте, так само як і відсоток тих автомобілей, що припливли до нас саме з США.

Найбільш популярним ціновим сегментом є діапазон від 17 до 35 тисяч доларів. Частка його перевищила 50%. Серед дорогих брендів, які купували українці, - Aston Martin DBS, Bentley Bentayga, Continental GT і Mulsanne; Rolls-Royce Cullinan і Phantom, Mercedes S-Class Maybach і інші. Як ми бачимо з таблиці 1.1 й загальна вартість пригнаних авто за рік стабільно росте.

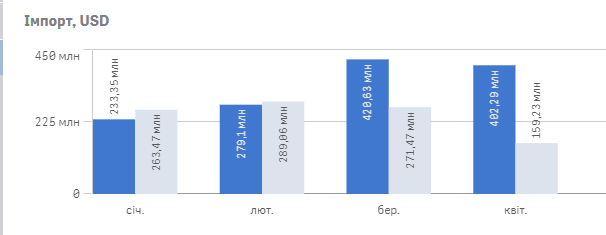
Як важливий факт, слід відзначити тенденцію зниження цін на вживані автомобілі. Очевидно, що це є наслідком збільшення кількості автомобілей на ринку. Адже лише за березень 2021р., в Україну було ввезено транспортних засобів на 420 млн. доларів. Що є більшою половиною від річної суми за цілий минулий рік рис. 1.2.

Рисунок 1.2 – Імпорт авто в Україну

*Джерело: [6]*

Всі ці фактори впливають на розповсюдження на перший погляд ексклюзивних авто.

# **Архітектура розроблюваного веб додатку**

## **Еталонна модель веб сайту**

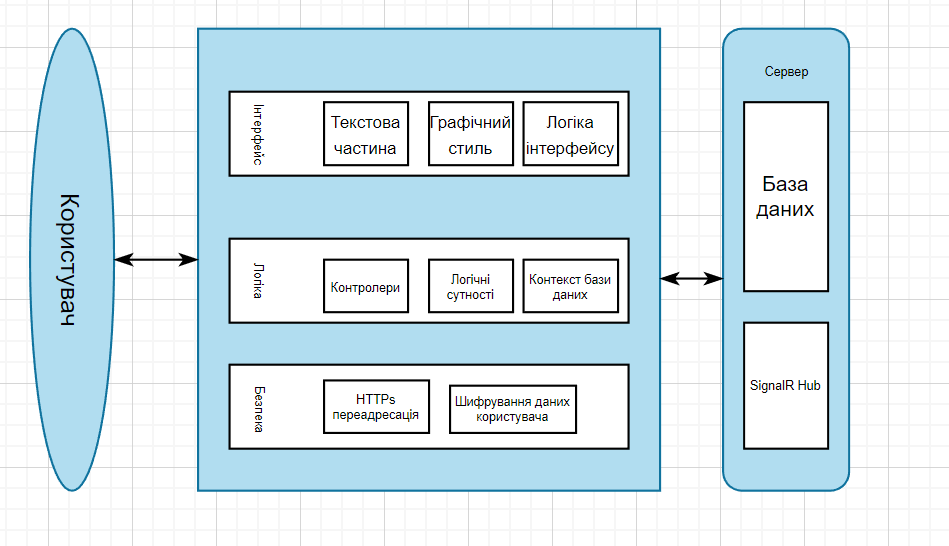


Рисунок 1. Еталонна модель веб сайту

*Джерело: побудовано автором*

## Блок схема веб сайту

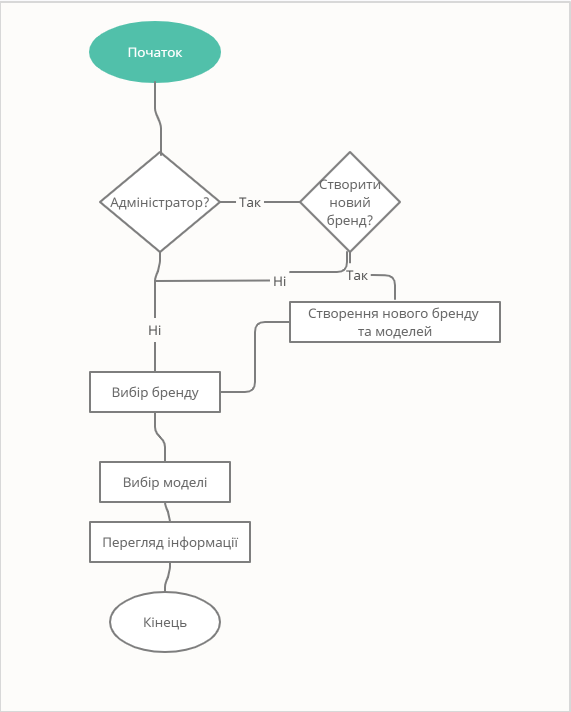


Рисунок 2. Блок схема веб сайту

*Джерело: побудовано автором*

## Use case діаграма

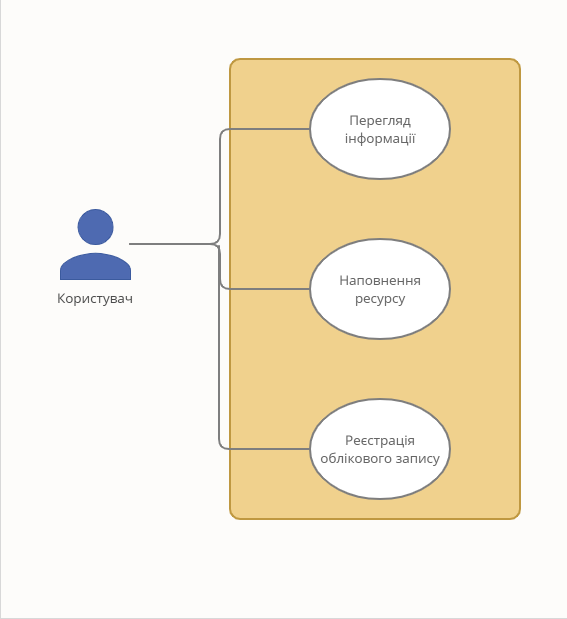


Рисунок 3. Use case діаграма

*Джерело: побудовано автором*

## Діаграма класів

Рисунок 4. Діаграма класів

*Джерело: побудовано автором*

Діаграма кооперації

Рисунок 5. Діаграма кооперації

*Джерело: побудовано автором*

## Діаграма MVC архітектури

Рисунок 6. Діаграма послідовності

*Джерело:*

## Діаграма стану

Рисунок 7. Діаграма стану

*Джерело: побудовано автором*

## Діаграма розгортання

Рисунок 8. Діаграма розгортання

*Джерело: побудовано автором*

# Опис бази даних

## Логічна модель бази даних

Рисунок 10. Логічна модель бази даних

*Джерело: побудовано автором*

## Фізична модель бази даних

## Нормалізація бази даних

## Прив’язка бази даних та веб додатку

Рисунок 11. Діаграма роботи з базою даних

*Джерело: побудовано автором (знімок екрану)*

## Скріншоти наповненої бази даних

Рисунок 12. Наповнена таблиця ContentUrls

*Джерело: побудовано автором (знімок екрану)*

Рисунок 13. Наповнена таблиця ContentInfos

*Джерело: побудовано автором (знімок екрану)*

# Дизайн проекту у Figma

Для побудови графічної частини програми, потрібно розробити дизайн, як найкраще для створення прототипу сайту підходить додаток Figma. Figma – це хмарний багатоплатформовий сервіс для дизайнерів інтерфейсів і web-розробників, з яким можна працювати безпосередньо в браузері. І це лише одне з важливих переваг платформи.



Рисунок 14. Дизайн основної сторінки

*Джерело: побудовано автором (знімок екрану)*

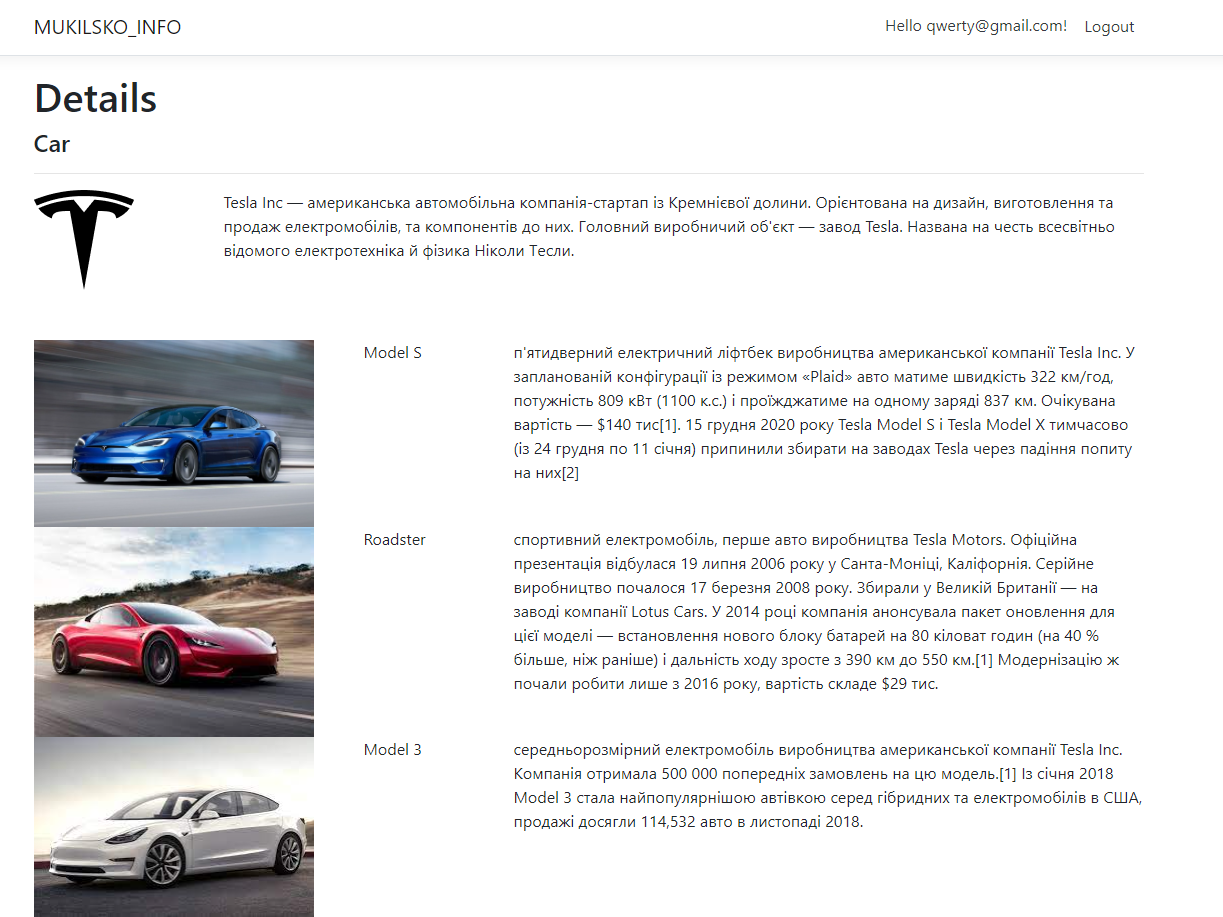


Рисунок 15. Модельний ряд

*Джерело: побудовано автором (знімок екрану)*

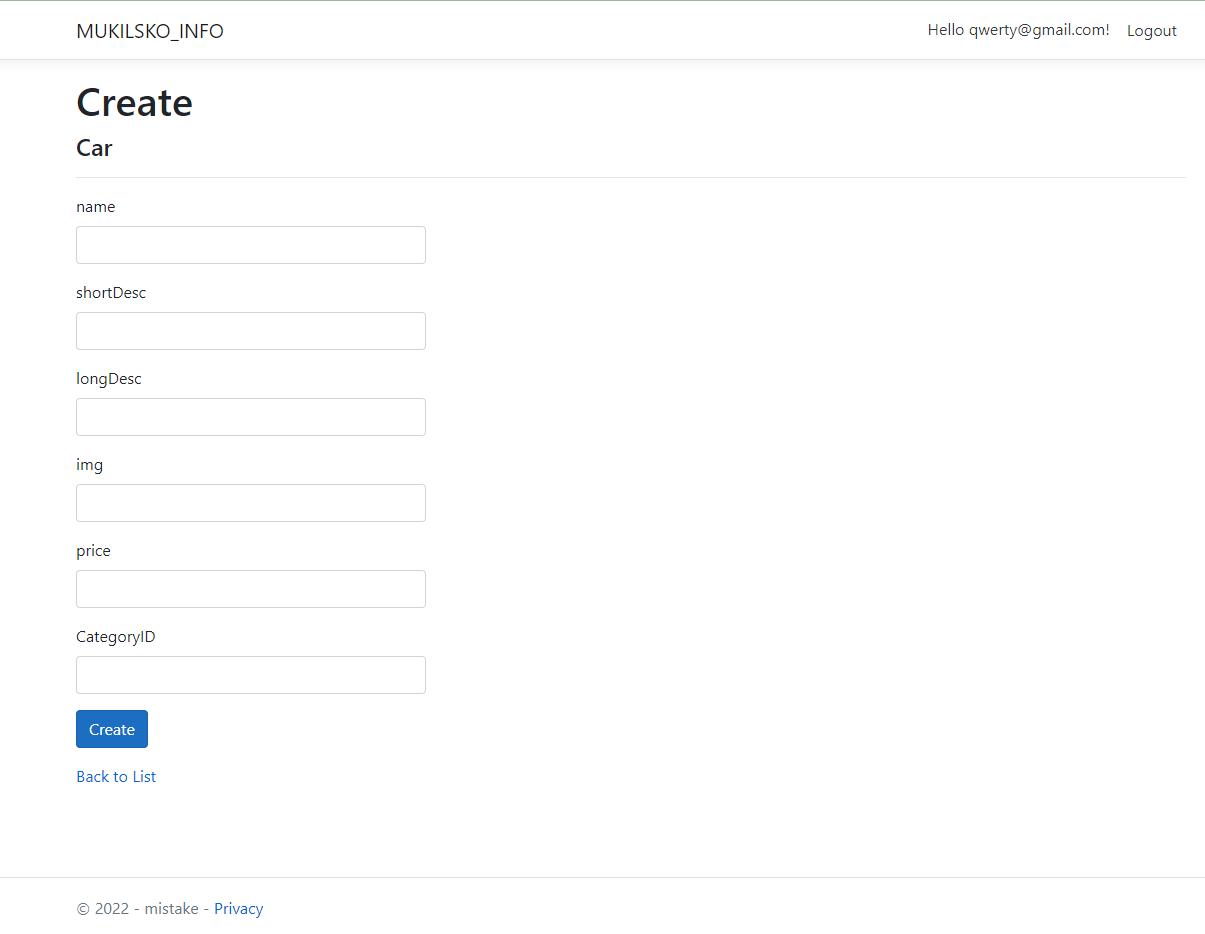


Рисунок 16. Створення Бренду

*Джерело: побудовано автором (знімок екрану)*

# Додатки

## Додаток А

Програмний код HomeController.cs

using System;

using Microsoft.AspNetCore.Mvc;

using Microsoft.Extensions.Logging;

using mistake.Models;

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.Linq;

using System.Threading.Tasks;

namespace mistake.Controllers

{

public class HomeController : Controller

{

private readonly ILogger<HomeController> \_logger;

public HomeController(ILogger<HomeController> logger)

{

\_logger = logger;

}

public IActionResult Index()

{

return View();

}

public IActionResult Privacy()

{

return View();

}

[ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]

public IActionResult Error()

{

return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });

}

}

}

## Додаток Б

Програмний код CarsController.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Authorization;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.Rendering;

using Microsoft.EntityFrameworkCore;

using mistake.Data;

using mistake.Models;

namespace mistake

{

[Authorize]

public class CarsController : Controller

{

private readonly ApplicationDbContext \_context;

public CarsController(ApplicationDbContext context)

{

\_context = context;

}

// GET: Cars

public async Task<IActionResult> Index()

{

return View(await \_context.Cars.ToListAsync());

}

// GET: Cars/Details/5

public async Task<IActionResult> Details(int? id)

{

if (id == null)

{

return NotFound();

}

var car = await \_context.Cars

.FirstOrDefaultAsync(m => m.id == id);

if (car == null)

{

return NotFound();

}

var models = await \_context.Models.Where(x => x.CarId == car.id).ToListAsync();

ViewBag.Models = models;

return View(car);

}

// GET: Cars/Create

public IActionResult Create()

{

return View();

}

// POST: Cars/Create

// To protect from overposting attacks, enable the specific properties you want to bind to.

// For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<IActionResult> Create([Bind("id,name,shortDesc,longDesc,img,price,CategoryID")] Car car)

{

if (ModelState.IsValid)

{

\_context.Add(car);

await \_context.SaveChangesAsync();

return RedirectToAction(nameof(Index));

}

return View(car);

}

// GET: Cars/Edit/5

public async Task<IActionResult> Edit(int? id)

{

if (id == null)

{

return NotFound();

}

var car = await \_context.Cars.FindAsync(id);

if (car == null)

{

return NotFound();

}

return View(car);

}

// POST: Cars/Edit/5

// To protect from overposting attacks, enable the specific properties you want to bind to.

// For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<IActionResult> Edit(int id, [Bind("id,name,shortDesc,longDesc,img,price,CategoryID")] Car car)

{

if (id != car.id)

{

return NotFound();

}

if (ModelState.IsValid)

{

try

{

\_context.Update(car);

await \_context.SaveChangesAsync();

}

catch (DbUpdateConcurrencyException)

{

if (!CarExists(car.id))

{

return NotFound();

}

else

{

throw;

}

}

return RedirectToAction(nameof(Index));

}

return View(car);

}

// GET: Cars/Delete/5

public async Task<IActionResult> Delete(int? id)

{

if (id == null)

{

return NotFound();

}

var car = await \_context.Cars

.FirstOrDefaultAsync(m => m.id == id);

if (car == null)

{

return NotFound();

}

return View(car);

}

// POST: Cars/Delete/5

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public async Task<IActionResult> DeleteConfirmed(int id)

{

var car = await \_context.Cars.FindAsync(id);

\_context.Cars.Remove(car);

await \_context.SaveChangesAsync();

return RedirectToAction(nameof(Index));

}

private bool CarExists(int id)

{

return \_context.Cars.Any(e => e.id == id);

}

}

}

## Додаток В

Програмний код ModelsController.cs

using System.Diagnostics;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.Rendering;

using Microsoft.EntityFrameworkCore;

using mistake.Data;

using mistake.Models;

namespace mistake

{

public class ModelsController : Controller

{

private readonly ApplicationDbContext \_context;

public ModelsController(ApplicationDbContext context)

{

\_context = context;

}

// GET: Models

public async Task<IActionResult> Index()

{

return View(await \_context.Models.ToListAsync());

}

// GET: Models/Details/5

public async Task<IActionResult> Details(int? id)

{

if (id == null)

{

return NotFound();

}

var model = await \_context.Models

.FirstOrDefaultAsync(m => m.Id == id);

if (model == null)

{

return NotFound();

}

return View(model);

}

// GET: Models/Create

public IActionResult Create()

{

return View();

}

// POST: Models/Create

// To protect from overposting attacks, enable the specific properties you want to bind to.

// For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<IActionResult> Create([Bind("Id,CarId,Name,Img,Details")] Model model)

{

if (ModelState.IsValid)

{

\_context.Add(model);

await \_context.SaveChangesAsync();

return RedirectToAction(nameof(Create));

}

return View(model);

}

// GET: Models/Edit/5

public async Task<IActionResult> Edit(int? id)

{

if (id == null)

{

return NotFound();

}

var model = await \_context.Models.FindAsync(id);

if (model == null)

{

return NotFound();

}

return View(model);

}

// POST: Models/Edit/5

// To protect from overposting attacks, enable the specific properties you want to bind to.

// For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<IActionResult> Edit(int id, [Bind("Id,CarId,Name,Img,Details")] Model model)

{

if (id != model.Id)

{

return NotFound();

}

if (ModelState.IsValid)

{

try

{

\_context.Update(model);

await \_context.SaveChangesAsync();

}

catch (DbUpdateConcurrencyException)

{

if (!ModelExists(model.Id))

{

return NotFound();

}

else

{

throw;

}

}

return RedirectToAction(nameof(Index));

}

return View(model);

}

// GET: Models/Delete/5

public async Task<IActionResult> Delete(int? id)

{

if (id == null)

{

return NotFound();

}

var model = await \_context.Models

.FirstOrDefaultAsync(m => m.Id == id);

if (model == null)

{

return NotFound();

}

return View(model);

}

// POST: Models/Delete/5

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public async Task<IActionResult> DeleteConfirmed(int id)

{

var model = await \_context.Models.FindAsync(id);

\_context.Models.Remove(model);

await \_context.SaveChangesAsync();

return RedirectToAction(nameof(Index));

}

private bool ModelExists(int id)

{

return \_context.Models.Any(e => e.Id == id);

}

}

}