# Web Page:

## HTML File Code:

<html>

<head>

<link href="Content/bootstrap.min.css" rel="stylesheet" />

<script src="Scripts/angular.min.js"></script>

<script src="Scripts/jquery-3.3.1.min.js"></script>

<script src="Scripts/bootstrap.min.js"></script>

<script src="Scripts/module.js"></script>

<script src="Scripts/PersonController.js"></script>

<script src="Scripts/service.js"></script>

</head>

<body ng-app="CarDashboardDashboard" ng-controller="myController">

<div class="container">

<h2>Car Record</h2>

<button type="button" class="btn btn-primary" data-toggle="modal" data-target="#myModal" id="btnEmpty" ng-click="displayForAdd()">Add New Record</button><br /><br />

<table class="table">

<thead class="thead-dark">

<tr>

<th scope="col">Id</th>

<th scope="col"> Name</th>

<th scope="col">Model</th>

<th scope="col">Service</th>

</tr>

</thead>

<tbody id="tabcontent">

<tr ng-repeat="x **in** data">

<td>**{{** x.Id **}}**</td>

<td>**{{** x.Name **}}**</td>

<td>**{{** x.Model **}}**</td>

<td>**{{** x.Service **}}**</td>

<td><a href="#" data-toggle="modal" data-target="#myModal" ng-click="displayForUpdate(x)">Edit</a>|<a href="#" data-toggle="modal" data-target="#myModal" ng-click="displayForRemove(x)">Delete</a></td>

</tr>

</tbody>

</table>

</div>

<div class="modal" id="myModal" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<h4 class="modal-title" id="myModalLabel">**{{**operation**}}**</h4>

<button type="button" class="close" data-dismiss="modal">×</button>

</div>

<div class="modal-body">

<form>

<div class="form-group">

<label for="EmployeeId">ID</label>

<input type="text" class="form-control" ng-model="Id" placeholder="Id" />

</div>

<div class="form-group">

<label for="Name">Name</label>

<input type="text" class="form-control" ng-model="Name" placeholder="Name" />

</div>

<div class="form-group">

<label for="Name">Model</label>

<input type="text" class="form-control" ng-model="Model" placeholder="Model" />

</div>

<div class="form-group">

<label for="Name">Service</label>

<input type="text" class="form-control" ng-model="Service" placeholder="Service" />

</div>

</div>

</div>

</form>

</div>

<div class="modal-footer">

<!-- <button type="button" class="btn btn-primary" id="btnAdd">Add</button> -->

<button type="button" class="btn btn-success" data-target="myModal" data-dismiss="modal" ng-click="addNewRecord()" id="btnAdd">Add</button>

<button type="button" class="btn btn-danger" data-bind="myModal" data-dismiss="modal" ng-click="deleteRecord()" id="btnDelete">Delete</button>

<button type="button" class="btn btn-primary" data-bind="myModal" data-dismiss="modal" ng-click="updateRecord()" id="btnUpdate">Update</button>

<button type="button" class="btn btn-default" data-dismiss="modal">Close</button>

</div>

</div>

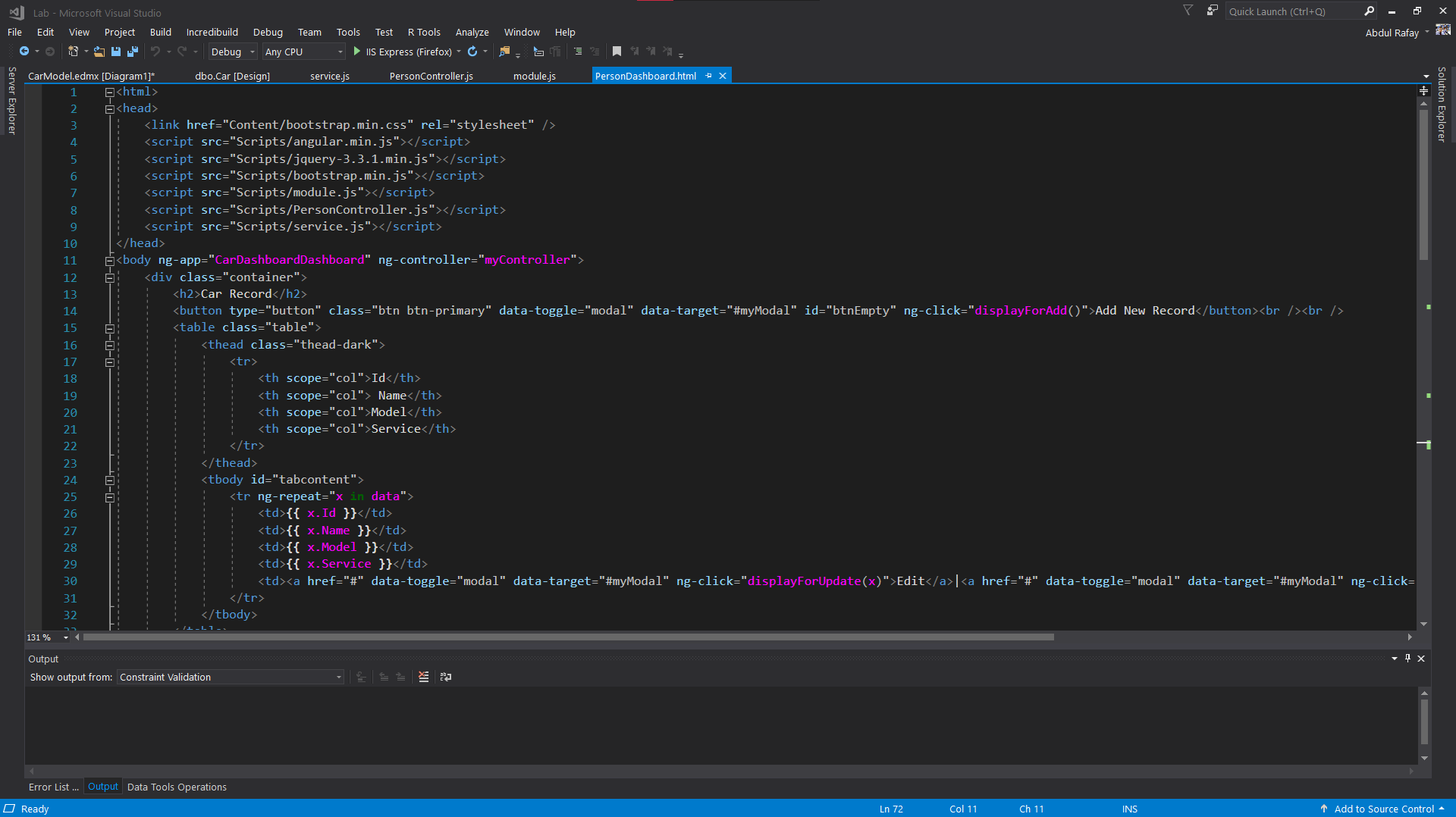
</div>

</div>

</body>

</html>

## HTML Code Screenshot:



## Webpage Controller:

/// <reference path="angular.min.js" />

/// <reference path="module.js" />

myApp.controller("myController", function ($scope, AppService) {

var result = AppService.get("/api/CarController");

result.then(function (response) {

alert(JSON.stringify(response.data));

$scope.data = response.data;

},

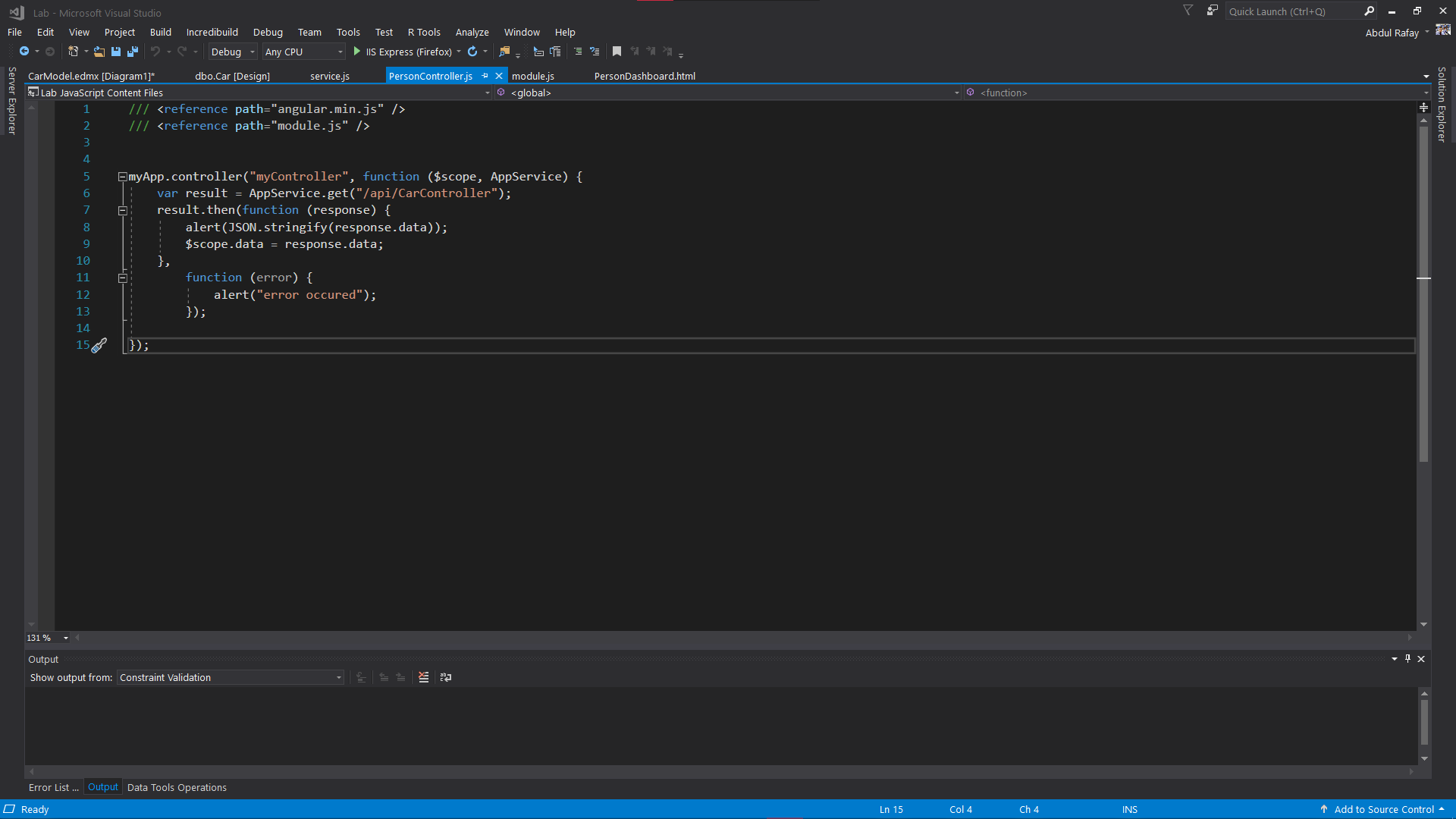
function (error) {

alert("error occured");

});

});

## Web Page Controller Screenshot:

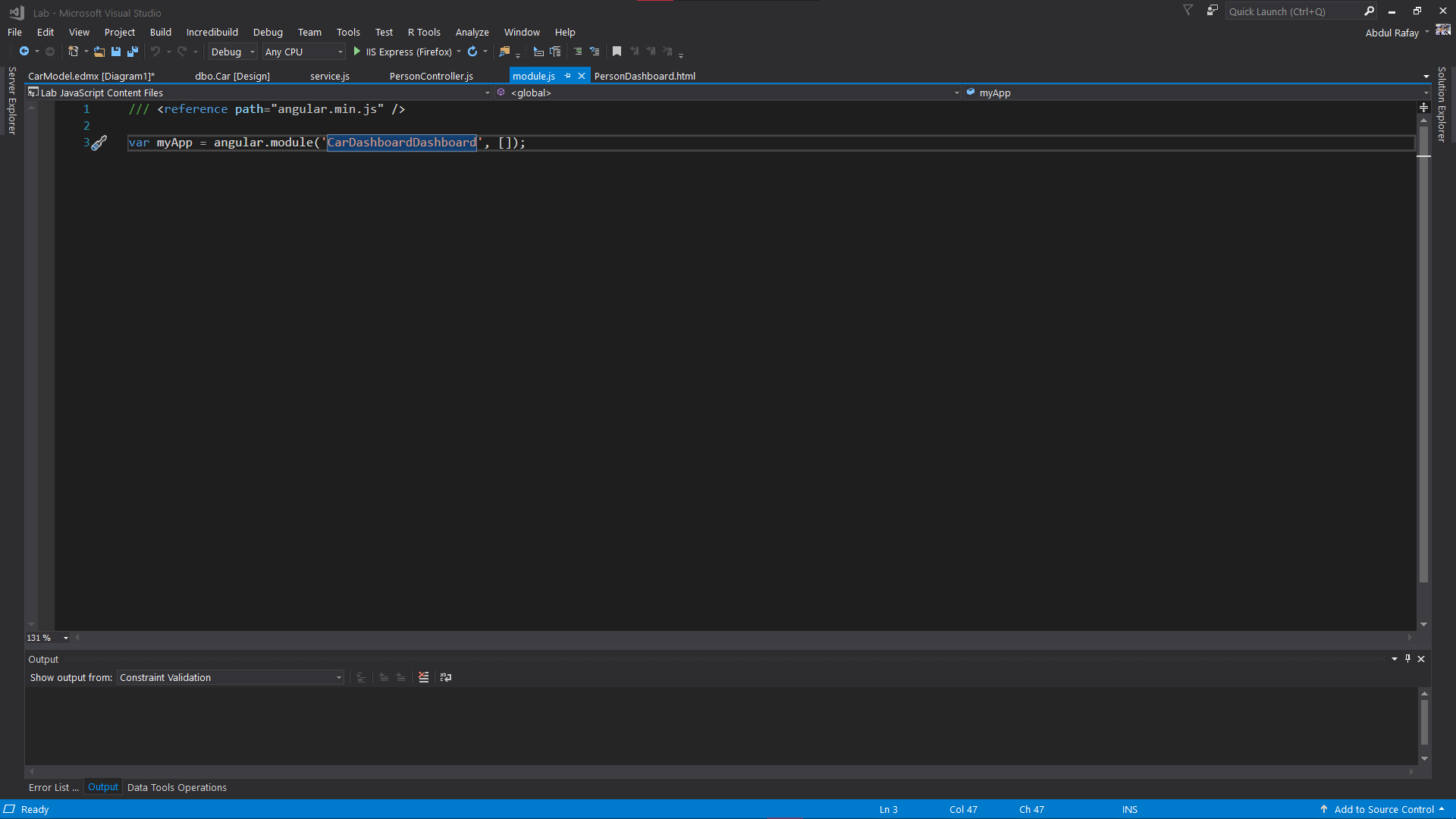


## Module.Js File:

/// <reference path="angular.min.js" />

var myApp = angular.module('CarDashboardDashboard', []);

## Module.JS Screenshot:



## Service.Js File:

/// <reference path="angular.min.js" />

/// <reference path="module.js" />

myApp.service("AppService", function ($http) {

this.post = function (data, url) {

var promise = $http.post(url, data);

return promise;

};

this.get = function (url) {

var promise = $http.get(url);

return promise;

};

});

mpApp.service("AppService", function ($http) {

this.push = function (data, url) {

var promise = $https.push(url, data);

return promise;

};

this.push = function (url) {

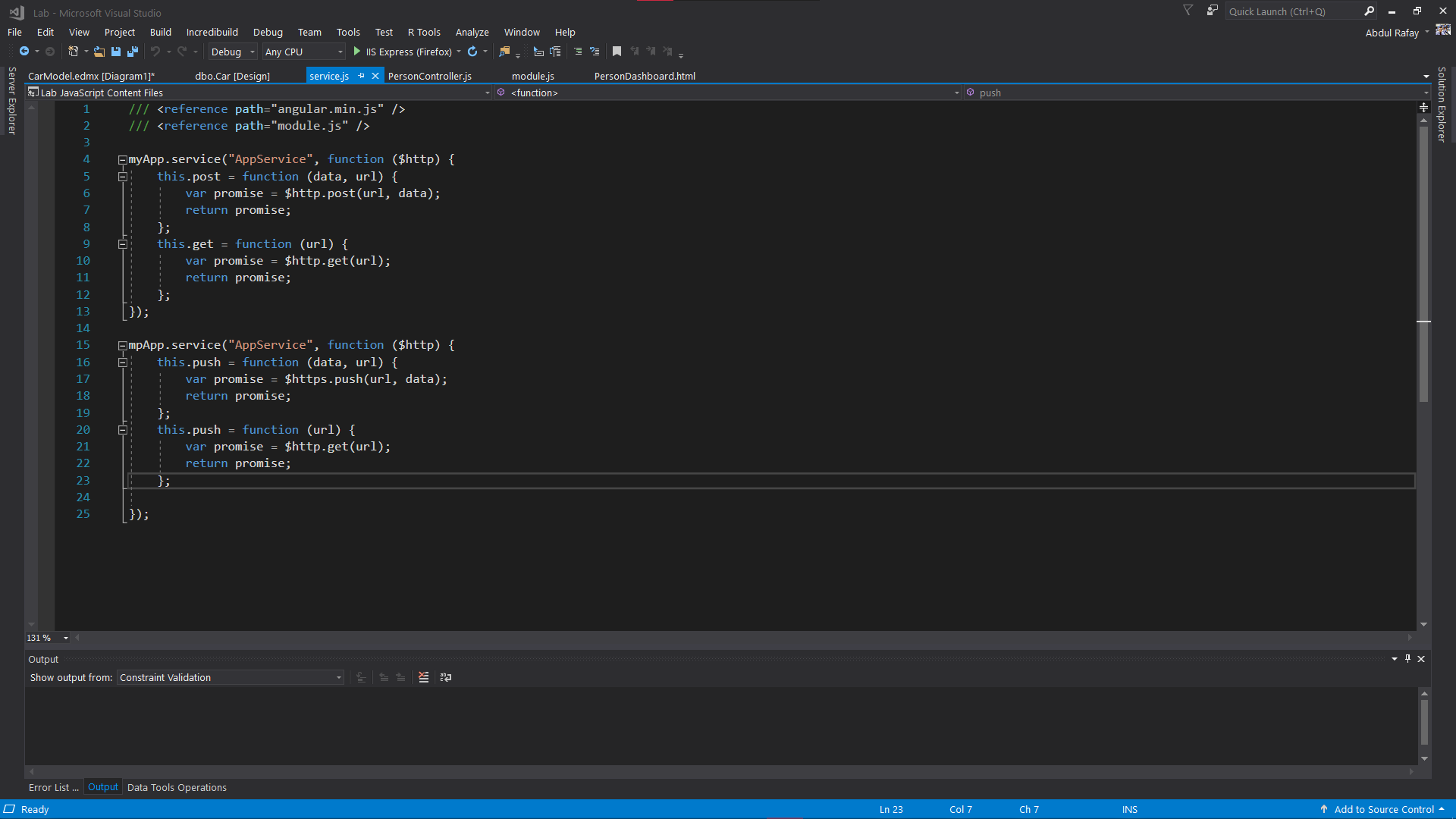
var promise = $http.get(url);

return promise;

};

});

## Service.Js Screenshots:



# Database:

## Table:

CREATE TABLE [dbo].[Car] (

[Id] INT IDENTITY (1, 1) NOT NULL,

[Name] TEXT NULL,

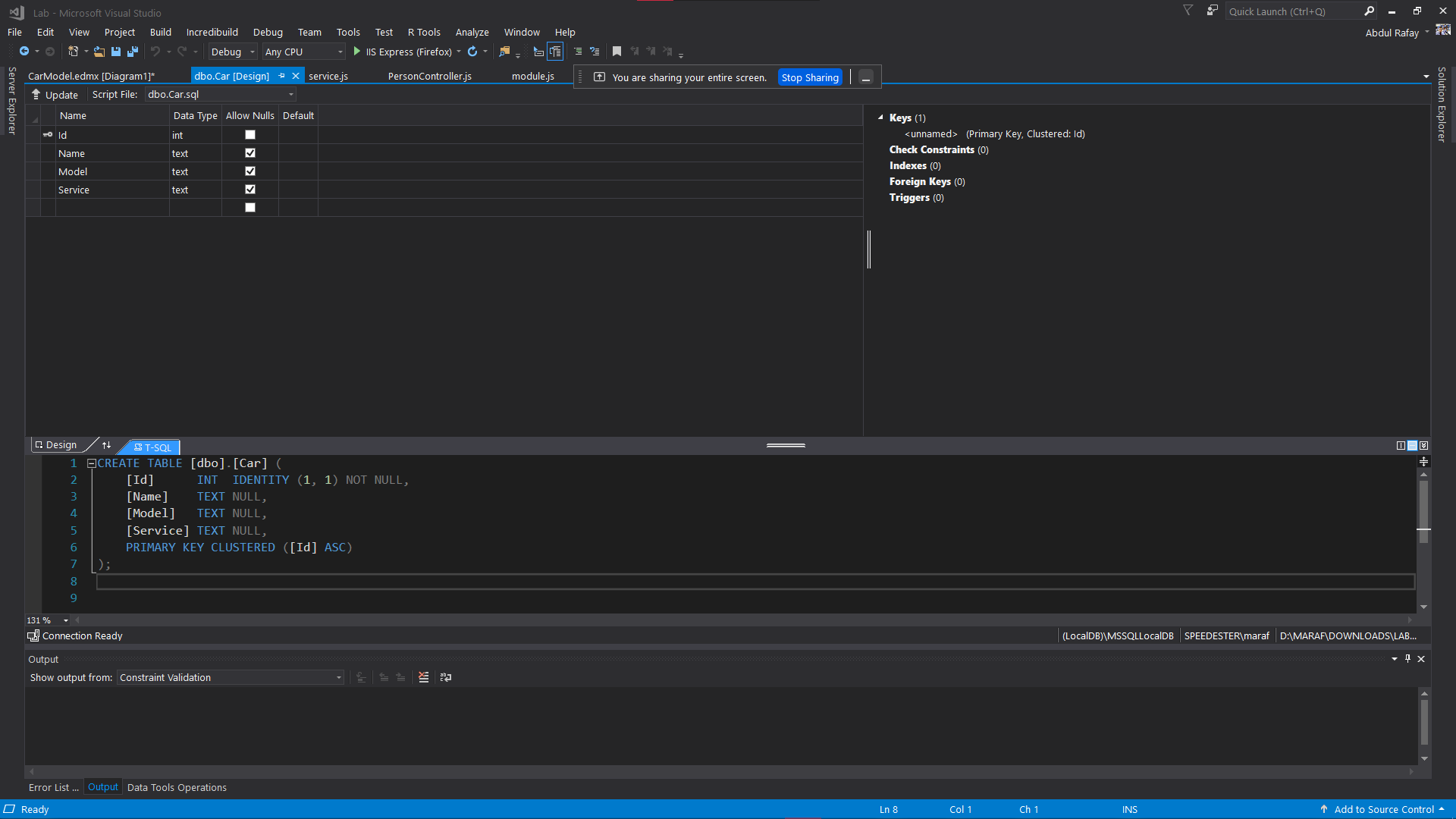
[Model] TEXT NULL,

[Service] TEXT NULL,

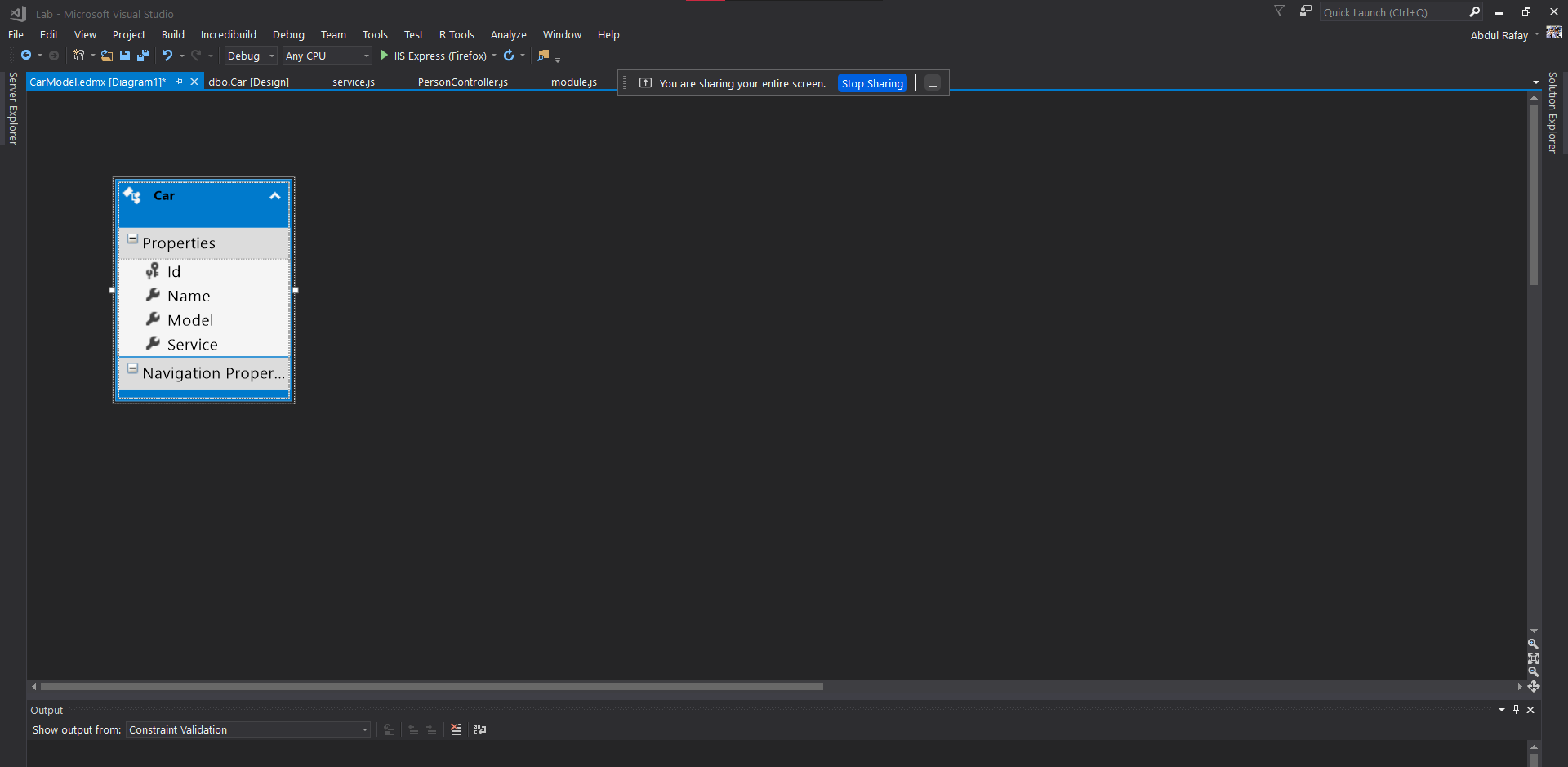
PRIMARY KEY CLUSTERED ([Id] ASC)

);

## Screenshot of the Table:



# ER Model:



# Car Table Controller:

## Code:

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Entity;

using System.Data.Entity.Infrastructure;

using System.Linq;

using System.Net;

using System.Net.Http;

using System.Web.Http;

using System.Web.Http.Description;

using Lab.Models;

namespace Lab.Controllers

{

public class CarsController : ApiController

{

private PersonDBEntities1 db = new PersonDBEntities1();

// GET: api/Cars

public IQueryable<Car> GetCars()

{

return db.Cars;

}

// GET: api/Cars/5

[ResponseType(typeof(Car))]

public IHttpActionResult GetCar(int id)

{

Car car = db.Cars.Find(id);

if (car == null)

{

return NotFound();

}

return Ok(car);

}

// PUT: api/Cars/5

[ResponseType(typeof(void))]

public IHttpActionResult PutCar(int id, Car car)

{

if (!ModelState.IsValid)

{

return BadRequest(ModelState);

}

if (id != car.Id)

{

return BadRequest();

}

db.Entry(car).State = EntityState.Modified;

try

{

db.SaveChanges();

}

catch (DbUpdateConcurrencyException)

{

if (!CarExists(id))

{

return NotFound();

}

else

{

throw;

}

}

return StatusCode(HttpStatusCode.NoContent);

}

// POST: api/Cars

[ResponseType(typeof(Car))]

public IHttpActionResult PostCar(Car car)

{

if (!ModelState.IsValid)

{

return BadRequest(ModelState);

}

db.Cars.Add(car);

db.SaveChanges();

return CreatedAtRoute("DefaultApi", new { id = car.Id }, car);

}

// DELETE: api/Cars/5

[ResponseType(typeof(Car))]

public IHttpActionResult DeleteCar(int id)

{

Car car = db.Cars.Find(id);

if (car == null)

{

return NotFound();

}

db.Cars.Remove(car);

db.SaveChanges();

return Ok(car);

}

protected override void Dispose(bool disposing)

{

if (disposing)

{

db.Dispose();

}

base.Dispose(disposing);

}

private bool CarExists(int id)

{

return db.Cars.Count(e => e.Id == id) > 0;

}

}

}

## Screenshot of Car table controller:

