AIRLINE RESERVATION SYSTEM

DOCUMENTATION

PROJECT GROUP MEMBER

STUDENT NAME	STUDENT ID
MUHAMMAD MOHTASHAM	STUDENT1431527
MIRZA USMAN ASAD BAIG	STUDENT1434945

CURRICULUM:7062

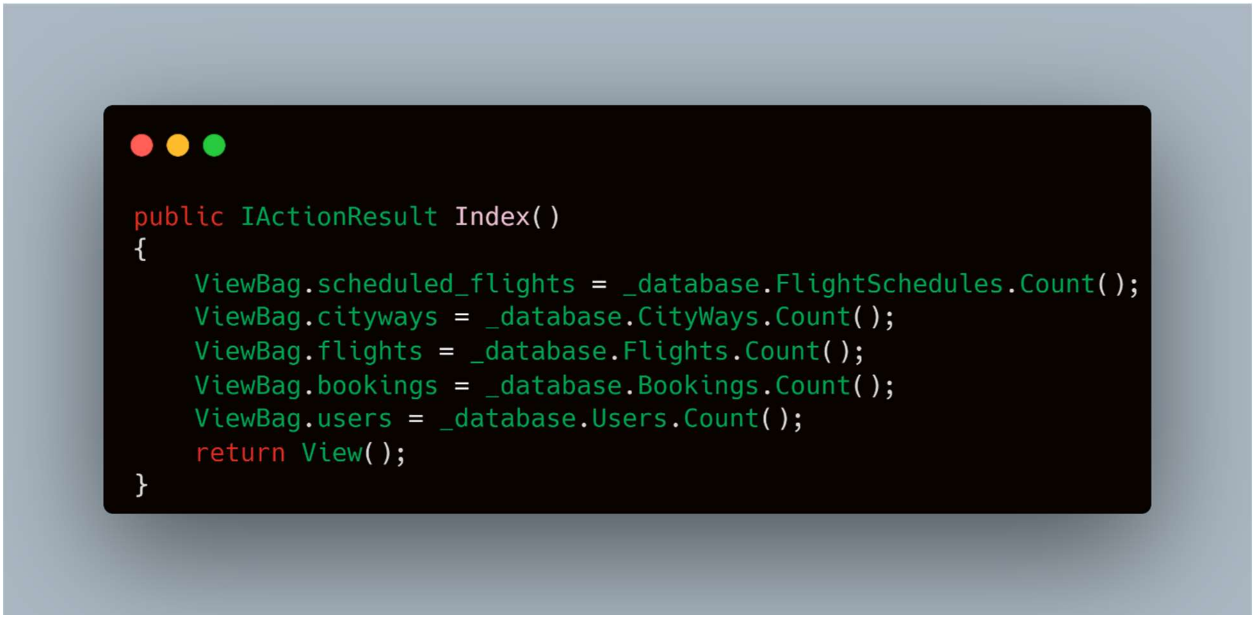
BATCH:2210G3

FACULTY:MISS BAKHTAWAR SIRAJ

CO-ORDINATOR:MISS HAFSA QADIR

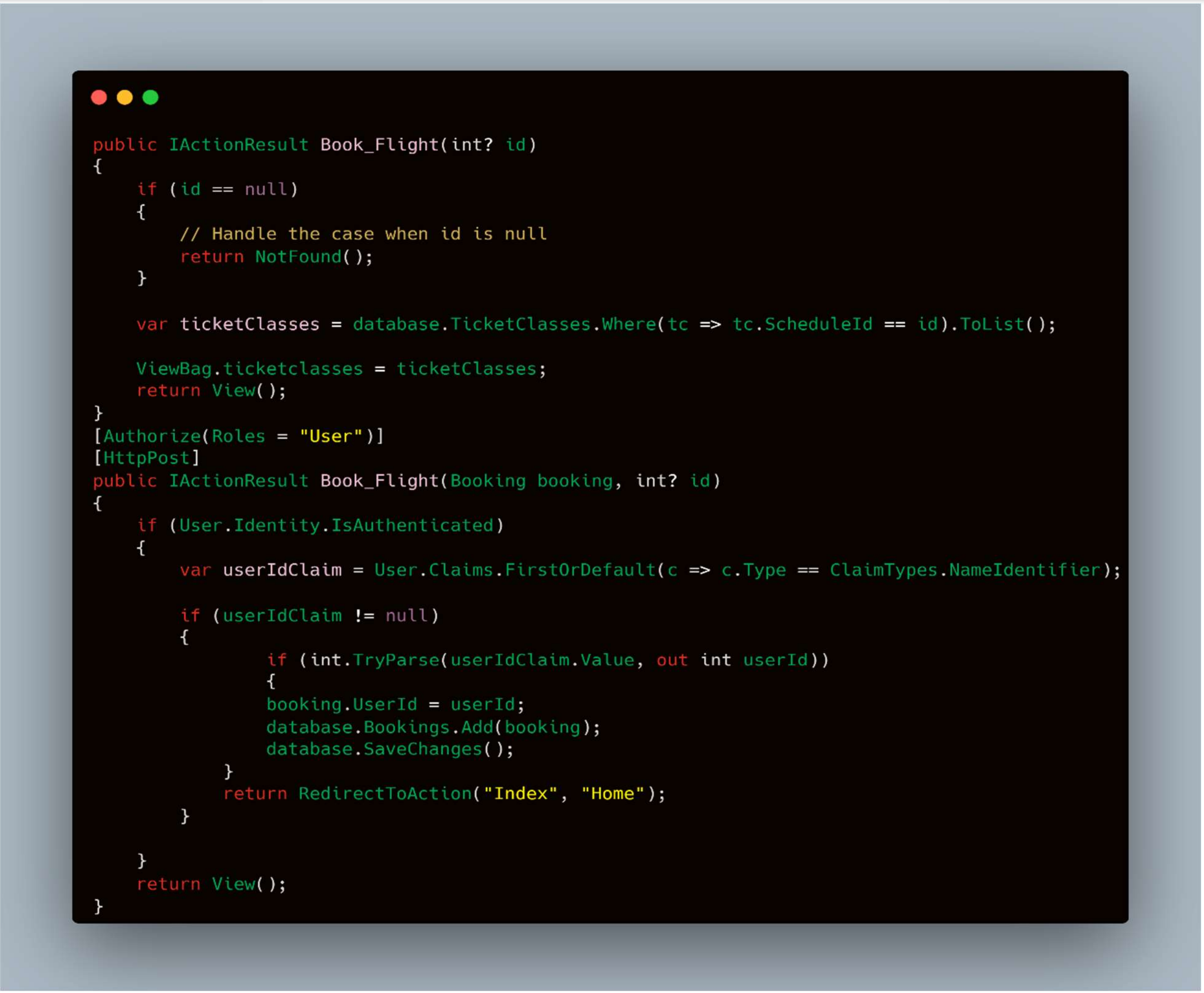
SCREEN SHOOTS

DEVELOPER GUIDE ADMIN DASHBOARD



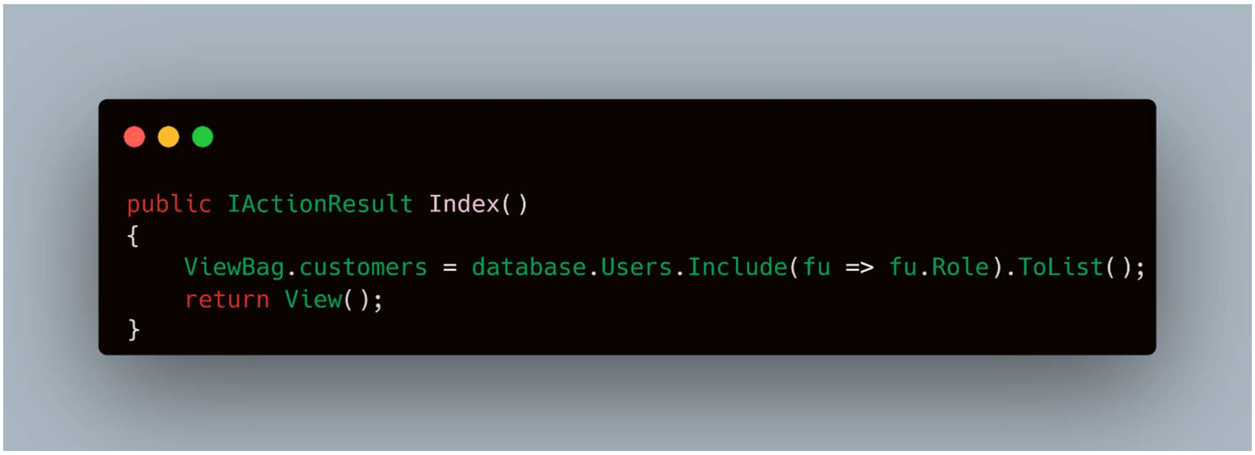
```
public IActionResult Index()
{
    ViewBag.scheduled_flights = _database.FlightSchedules.Count();
    ViewBag.cityways = _database.CityWays.Count();
    ViewBag.flights = _database.Flights.Count();
    ViewBag.bookings = _database.Bookings.Count();
    ViewBag.users = _database.Users.Count();
    return View();
}
```

BOOK FLIGHT



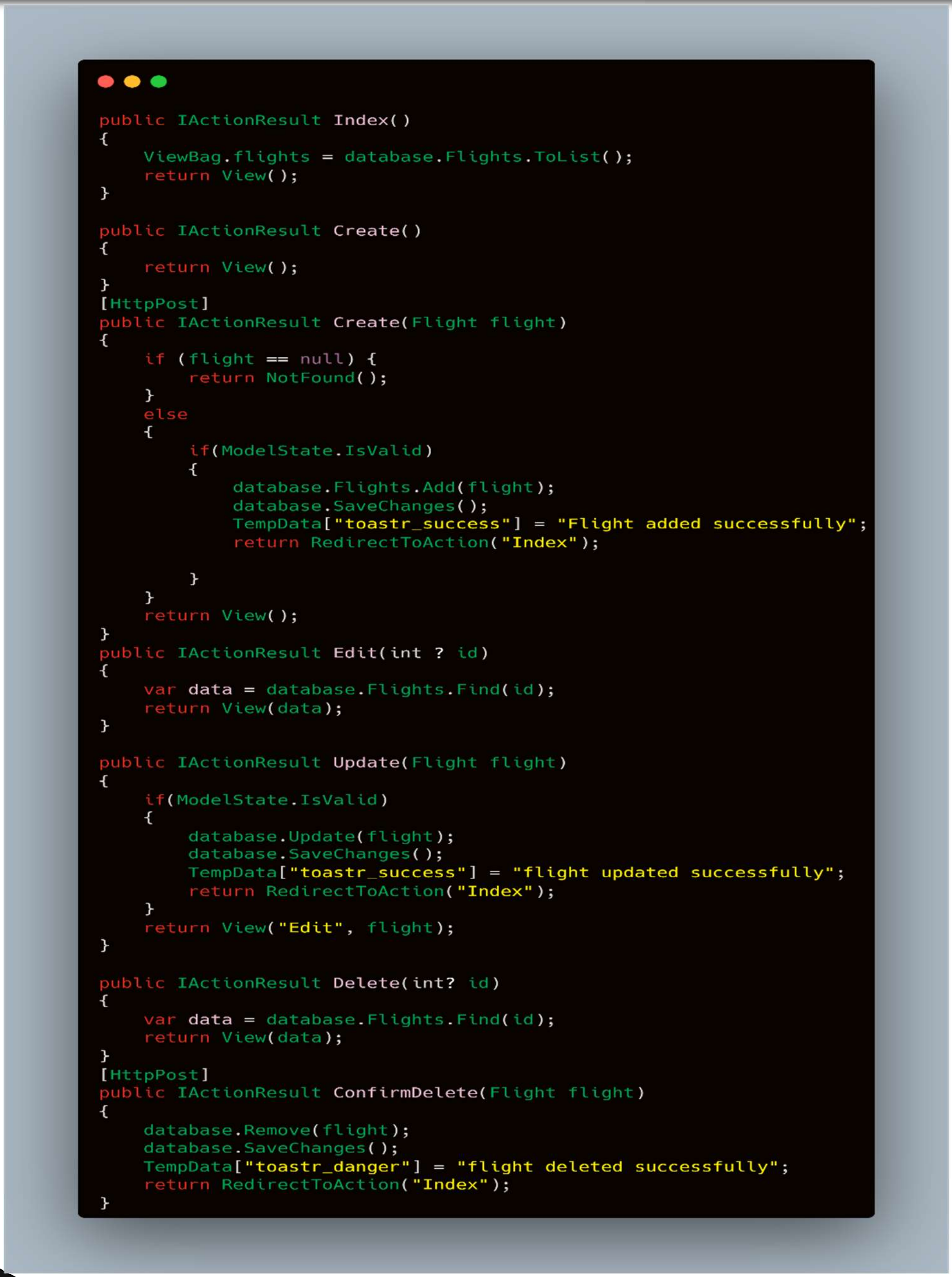
```
public IActionResult Book_Flight(int? id)
{
    if (id == null)
    {
        // Handle the case when id is null
        return NotFound();
    }

    var ticketClasses = database.TicketClasses.Where(tc => tc.ScheduleId == id).ToList();

    ViewBag.ticketclasses = ticketClasses;
    return View();
}
[Authorize(Roles = "User")]
[HttpPost]
public IActionResult Book_Flight(Booking booking, int? id)
{
    if (User.Identity.IsAuthenticated)
    {
        var userIdClaim = User.Claims.FirstOrDefault(c => c.Type == ClaimTypes.NameIdentifier);

        if (userIdClaim != null)
        {
                if (int.TryParse(userIdClaim.Value, out int userId))
                {
                booking.UserId = userId;
                database.Bookings.Add(booking);
                database.SaveChanges();
            }
            return RedirectToAction("Index", "Home");
        }

    }
    return View();
}
```

CUSTOMERS



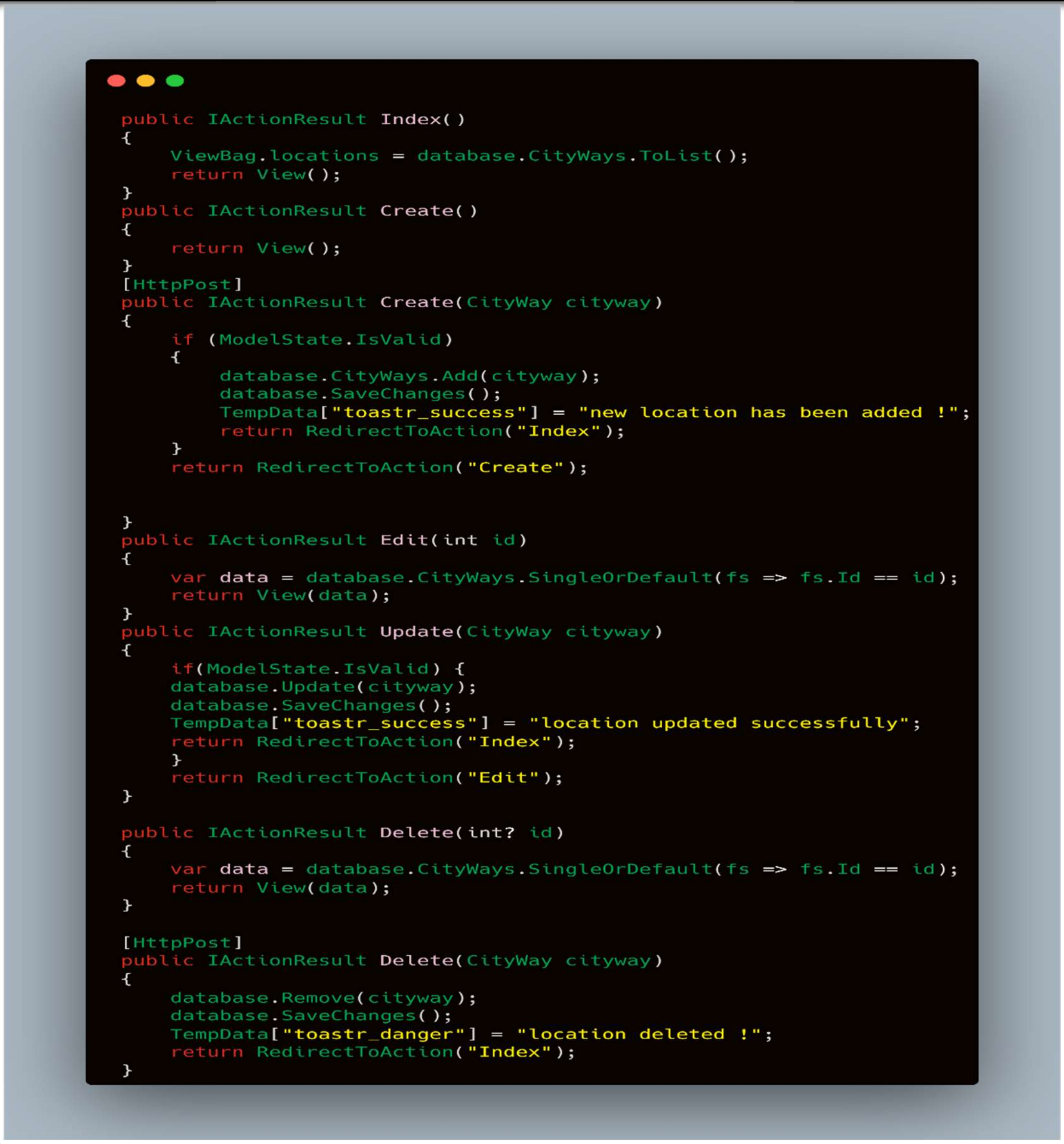
```
public IActionResult Index()
{
    ViewBag.customers = database.Users.Include(fu => fu.Role).ToList();
    return View();
}
```

FLIGHTS



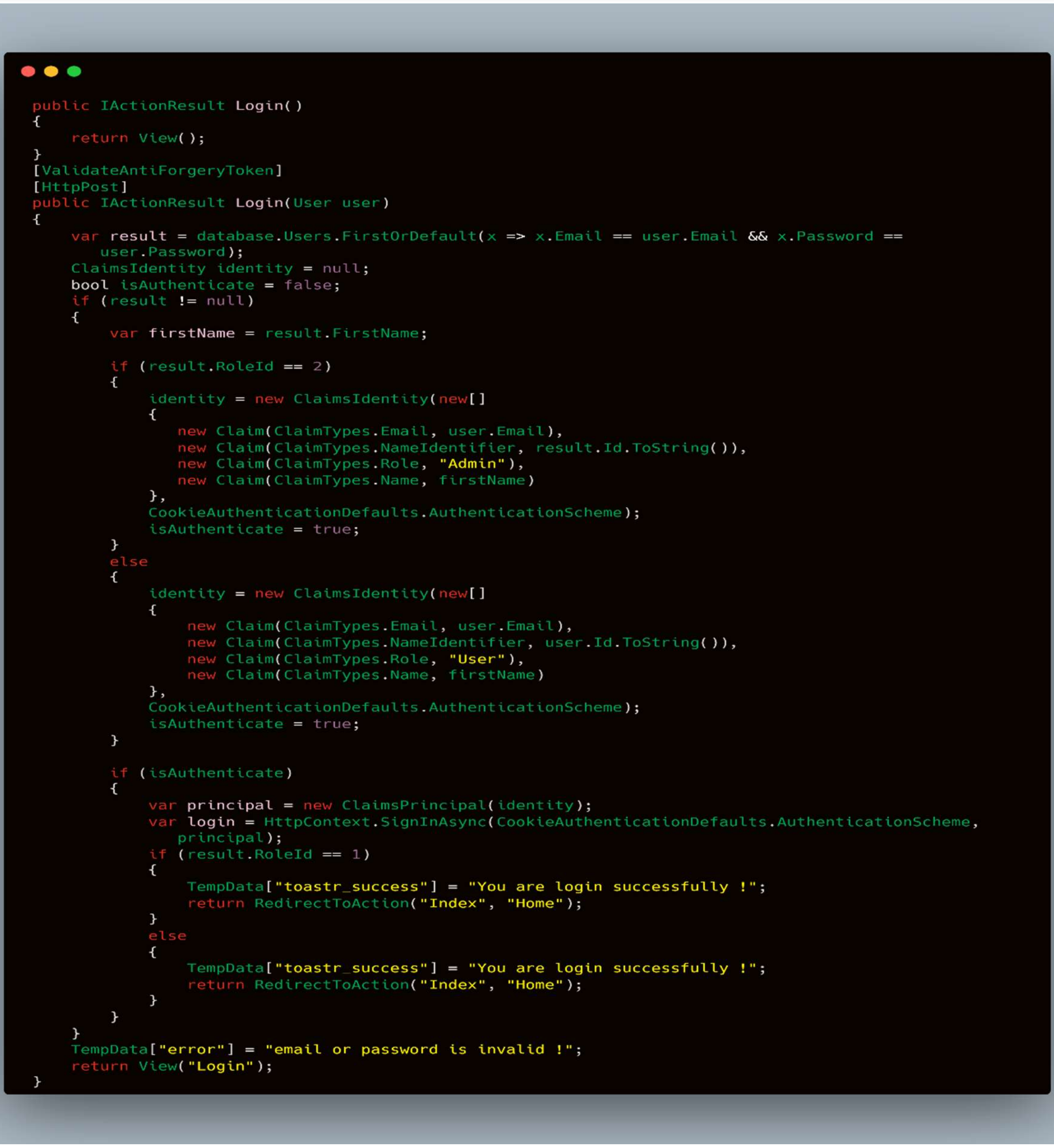
```
public IActionResult Index()
{
    ViewBag.flights = database.Flights.ToList();
    return View();
}

public IActionResult Create()
{
    return View();
}
[HttpPost]
public IActionResult Create(Flight flight)
{
    if (flight == null) {
        return NotFound();
    }
    else
    {
        if(ModelState.IsValid)
        {
            database.Flights.Add(flight);
            database.SaveChanges();
            TempData["toastr_success"] = "Flight added successfully";
            return RedirectToAction("Index");

        }
    }
    return View();
}
public IActionResult Edit(int ? id)
{
    var data = database.Flights.Find(id);
    return View(data);
}

public IActionResult Update(Flight flight)
{
    if(ModelState.IsValid)
    {
        database.Update(flight);
        database.SaveChanges();
        TempData["toastr_success"] = "flight updated successfully";
        return RedirectToAction("Index");
    }
    return View("Edit", flight);
}

public IActionResult Delete(int? id)
{
    var data = database.Flights.Find(id);
    return View(data);
}
[HttpPost]
public IActionResult ConfirmDelete(Flight flight)
{
    database.Remove(flight);
    database.SaveChanges();
    TempData["toastr_danger"] = "flight deleted successfully";
    return RedirectToAction("Index");
}
```

LOCATION



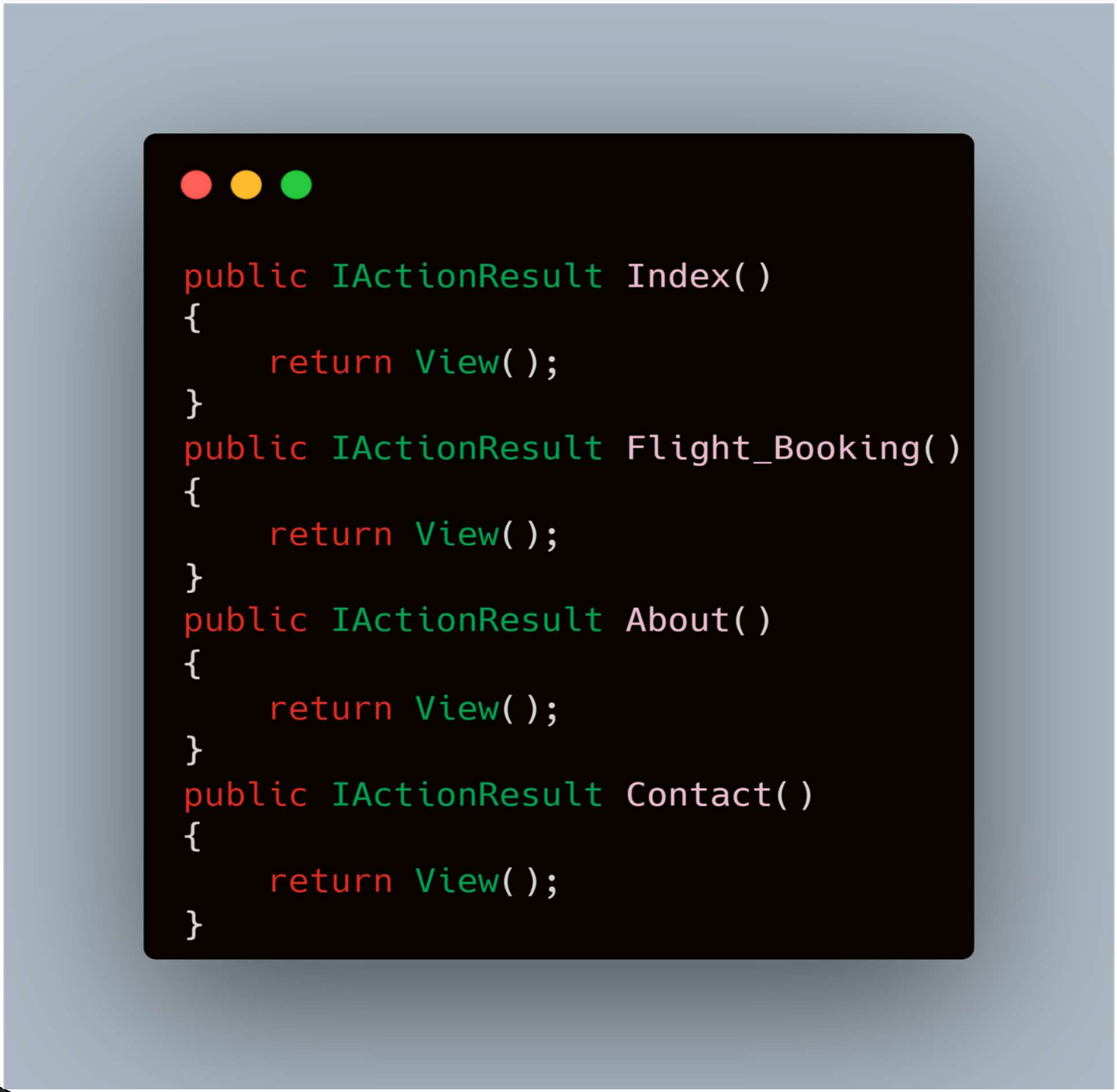
```
public IActionResult Index()
{
    ViewBag.locations = database.CityWays.ToList();
    return View();
}
public IActionResult Create()
{
    return View();
}
[HttpPost]
public IActionResult Create(CityWay cityway)
{
    if (ModelState.IsValid)
    {
        database.CityWays.Add(cityway);
        database.SaveChanges();
        TempData["toastr_success"] = "new location has been added !";
        return RedirectToAction("Index");
    }
    return RedirectToAction("Create");


}
public IActionResult Edit(int id)
{
    var data = database.CityWays.SingleOrDefault(fs => fs.Id == id);
    return View(data);
}
public IActionResult Update(CityWay cityway)
{
    if(ModelState.IsValid) {
    database.Update(cityway);
    database.SaveChanges();
    TempData["toastr_success"] = "location updated successfully";
    return RedirectToAction("Index");
    }
    return RedirectToAction("Edit");
}

public IActionResult Delete(int? id)
{
    var data = database.CityWays.SingleOrDefault(fs => fs.Id == id);
    return View(data);
}

[HttpPost]
public IActionResult Delete(CityWay cityway)
{
    database.Remove(cityway);
    database.SaveChanges();
    TempData["toastr_danger"] = "location deleted !";
    return RedirectToAction("Index");
}
```

LOGIN



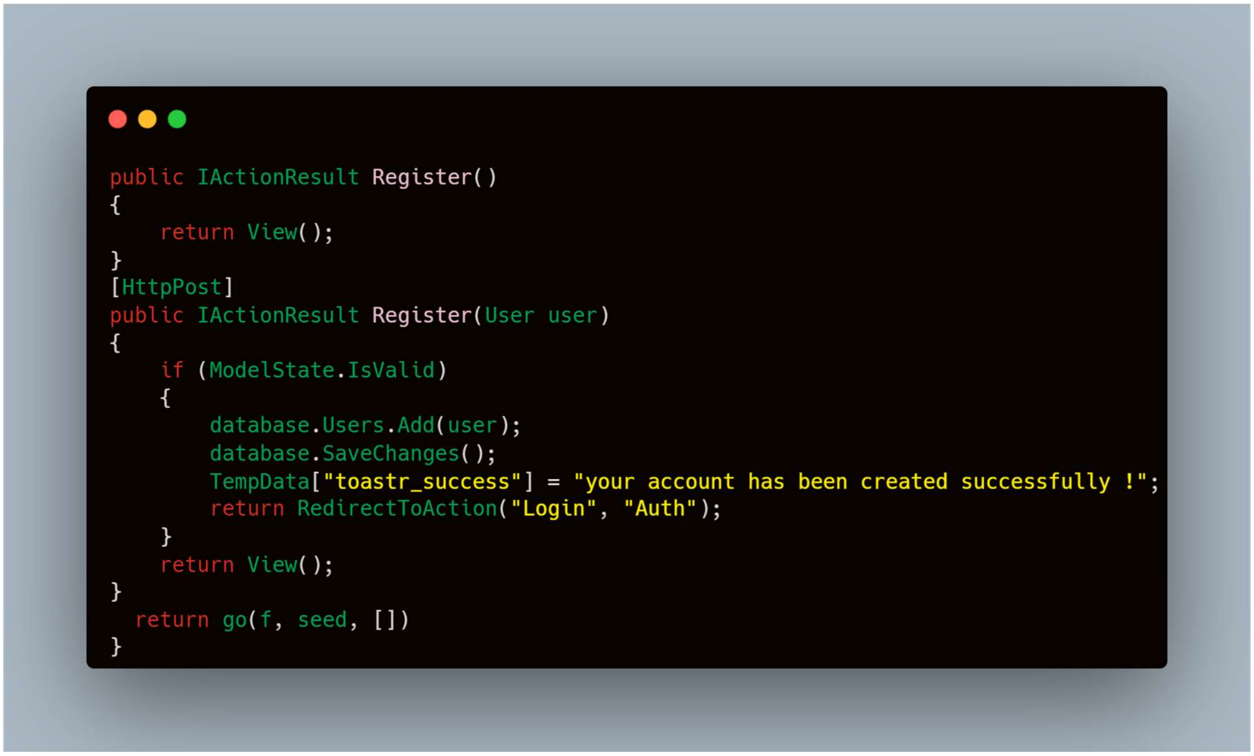
```
public IActionResult Login()
{
    return View();
}
[ValidateAntiForgeryToken]
[HttpPost]
public IActionResult Login(User user)
{
    var result = database.Users.FirstOrDefault(x => x.Email == user.Email && x.Password ==
       user.Password);
    ClaimsIdentity identity = null;
    bool isAuthenticate = false;
    if (result != null)
    {
        var firstName = result.FirstName;

        if (result.RoleId == 2)
        {
            identity = new ClaimsIdentity(new[]
            {
               new Claim(ClaimTypes.Email, user.Email),
               new Claim(ClaimTypes.NameIdentifier, result.Id.ToString()),
               new Claim(ClaimTypes.Role, "Admin"),
               new Claim(ClaimTypes.Name, firstName)
            },
            CookieAuthenticationDefaults.AuthenticationScheme);
            isAuthenticate = true;
        }
        else
        {
            identity = new ClaimsIdentity(new[]
            {
                new Claim(ClaimTypes.Email, user.Email),
                new Claim(ClaimTypes.NameIdentifier, user.Id.ToString()),
                new Claim(ClaimTypes.Role, "User"),
                new Claim(ClaimTypes.Name, firstName)
            },
            CookieAuthenticationDefaults.AuthenticationScheme);
            isAuthenticate = true;
        }

        if (isAuthenticate)
        {
            var principal = new ClaimsPrincipal(identity);
            var login = HttpContext.SignInAsync(CookieAuthenticationDefaults.AuthenticationScheme,
               principal);
            if (result.RoleId == 1)
            {
                TempData["toastr_success"] = "You are login successfully !";
                return RedirectToAction("Index", "Home");
            }
            else
            {
                TempData["toastr_success"] = "You are login successfully !";
                return RedirectToAction("Index", "Home");
            }
        }
    }
    TempData["error"] = "email or password is invalid !";
    return View("Login");
}
```

OTHER PAGES



```
public IActionResult Index()
{
    return View();
}
public IActionResult Flight_Booking()
{
    return View();
}
public IActionResult About()
{
    return View();
}
public IActionResult Contact()
{
    return View();
}
```

REGISTRATION FORM



```
public IActionResult Register()
{
    return View();
}
[HttpPost]
public IActionResult Register(User user)
{
    if (ModelState.IsValid)
    {
        database.Users.Add(user);
        database.SaveChanges();
        TempData["toastr_success"] = "your account has been created successfully !";
        return RedirectToAction("Login", "Auth");
    }
    return View();
}
  return go(f, seed, [])
}
```