QUEZON CITY UNIVERSITY

COLLEGE OF COMPUTER STUDIES
INFORMATION TECHNOLOGY

IPT102 – INTERMEDIATE PROGRAMMING AND TECHNOLOGIES 2

LABORATORY ACTIVITY 3

NAME:

SCORE	PERCENTAGE

CAHILIG, EFREN DAVE G. and LUMAGUE, JHAYRON U.

STUDENT NO: 23-2396 and 23-2335

YEAR/SECTION: SBIT3R

DATE: SEPTEMBER 9, 2025

INSTRUCTIONS:

Laboratory Activity: Introduction to MVC Architecture & Routing in ASP.NET Learning Objectives

Learning Objectives:

By the end of this activity, students will be able to:

1. Understand the Model-View-Controller (MVC) architecture in ASP.NET. 2. Create a simple ASP.NET MVC application.
3. Configure and test custom routes.
4. Display information using a controller, view, and model.

Activity Instructions:

Step 1: Create a New ASP.NET MVC Project

1. Open Visual Studio 2022 (or later).
2. Click Create a new project → Select ASP.NET Web Application (.NET Framework).
3. Name it: MvcRoutingDemo.
4. Choose MVC Template and click Create.

Step 2: Create a Model

1. Right-click on the Models folder → Add → Class.
2. Name it Student.cs.
3. Add the following code:

```
namespace MvcRoutingDemo.Models
{
    public class Student
    {
        public int Id { get; set; }
        public string FullName { get; set; }
        public string Course { get; set; }
    }
}
```

Step 3: Create a Controller

1. Right-click on the Controllers folder → Add → Controller.

2. Choose MVC 5 Controller – Empty → Name it StudentController.
3. Add the following code:


```
using System.Web.Mvc;
using MvcRoutingDemo.Models;

namespace MvcRoutingDemo.Controllers
{
   public class StudentController : Controller
   {
      // Default Action
      public ActionResult Index()
      {
         Student student = new Student()
         {
            Id = 1,
            FullName = "Juan Dela Cruz",
            Course = "BS Information Technology"
         };

         return View(student);
      }

      // Custom Route Example
      public ActionResult Details(int id)
      {
         Student student = new Student()
         {
            Id = id,
            FullName = "Student " + id,
            Course = "Sample Course"
         };

         return View(student);
      }
   }
}
```

Step 4: Create Views

1. Right-click on Index() → Add View → Template: Empty (without model).
2. Add this code in Index.cshtml:

QUEZON CITY UNIVERSITY
COLLEGE OF COMPUTER STUDIES
INFORMATION TECHNOLOGY

```
@model MvcRoutingDemo.Models.Student


<h2>Student Information</h2>
<p><b>ID:</b> @Model.Id</p>
<p><b>Name:</b> @Model.FullName</p>
<p><b>Course:</b> @Model.Course</p>
```

Do the same for Details(int id) → Add View Details.cshtml:

```
@model MvcRoutingDemo.Models.Student


<h2>Student Details</h2>
<p><b>ID:</b> @Model.Id</p>
<p><b>Name:</b> @Model.FullName</p>
<p><b>Course:</b> @Model.Course</p>
```

Step 5: Configure Routing

1. Open App_Start → RouteConfig.cs.
2. Update it as follows:

```
routes.MapRoute(
   name: "StudentDetails",
   url: "student/{id}",
   defaults: new { controller = "Student", action = "Details", id = UrlParameter.Optional }
);

routes.MapRoute(
   name: "Default",
   url: "{controller}/{action}/{id}",
   defaults: new { controller = "Student", action = "Index", id = UrlParameter.Optional }
);
```

Step 6: Run the Application

- Press F5 or Ctrl+F5 to run.
- Open in browser:

o http://localhost:xxxx/student → Shows default student info. o http://localhost:xxxx/student/5 → Shows student details with ID=5.

Activity Output

- Students will see a webpage displaying student information.
- They will test custom routing by changing the student ID in the URL.

Task for Students

1. Modify the model to include Age and Year Level.
2. Update both Index and Details views to display these new fields.
3. Create another route:
   - student/course/{courseName} → Displays a student with that course.