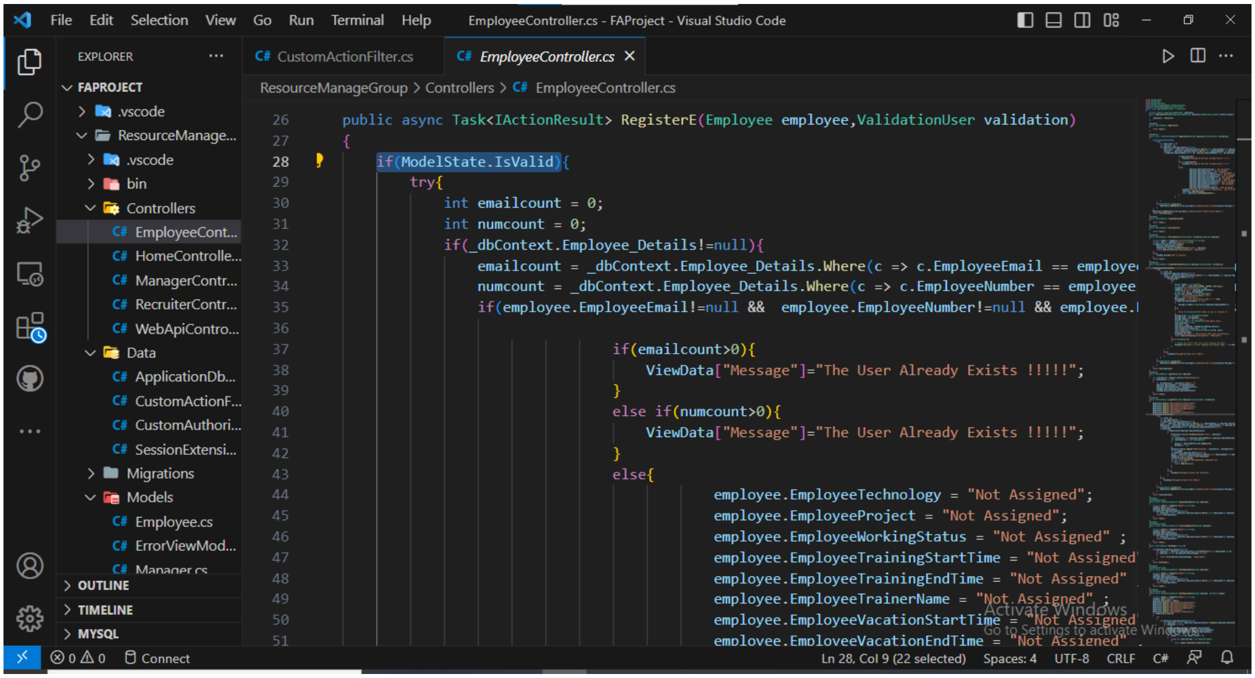
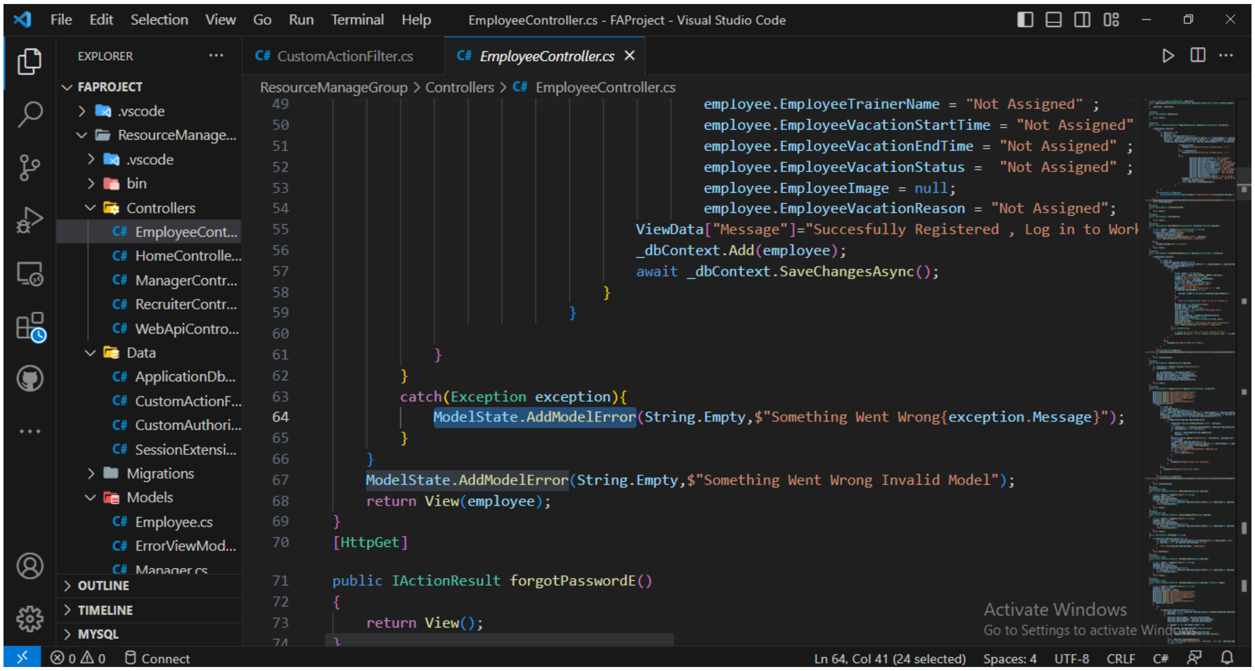
TASK - 7

Model State Error:

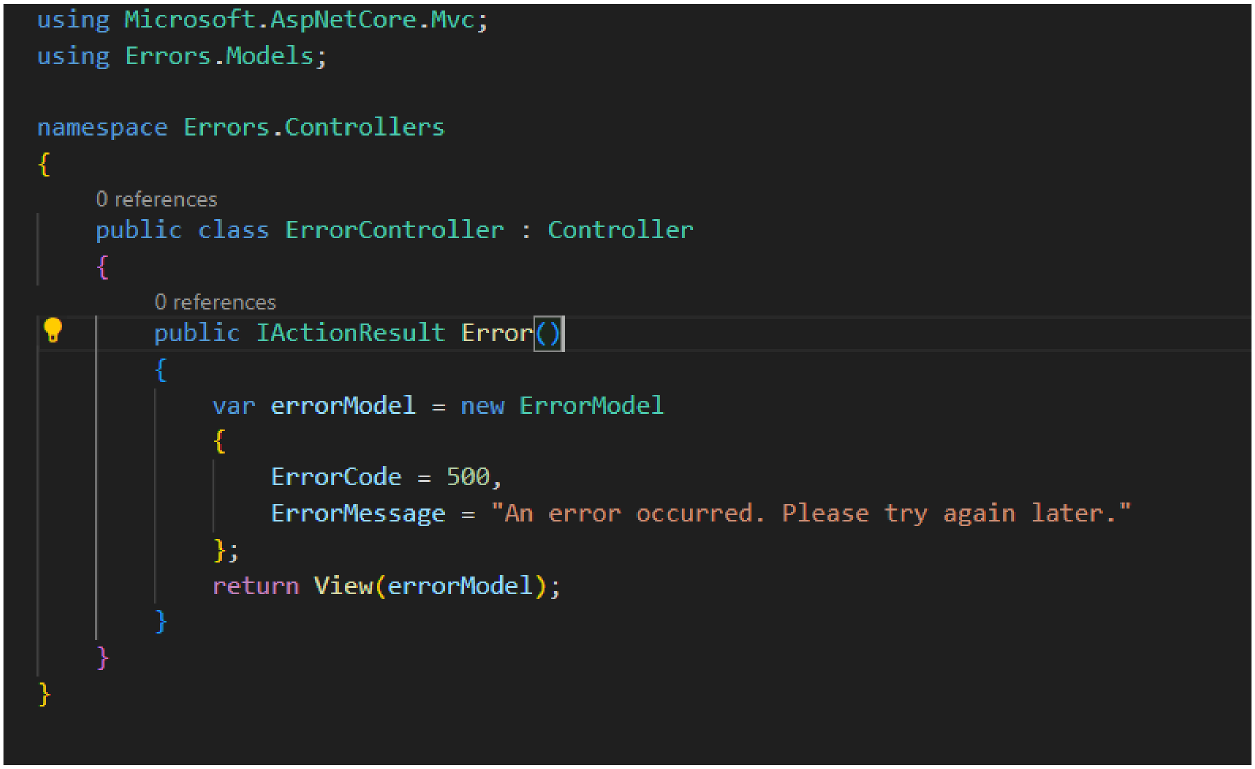


```
public async Task<IActionResult> RegisterE(Employee employee,ValidationUser validation)
{
    if(ModelState.IsValid){
        try{
            int emailcount = 0;
            int numcount = 0;
            if(_dbContext.Employee_Details!=null){
                emailcount = _dbContext.Employee_Details.Where(c => c.EmployeeEmail == employe
                numcount = _dbContext.Employee_Details.Where(c => c.EmployeeNumber == employee
                if(employee.EmployeeEmail!=null &&  employee.EmployeeNumber!=null && employee.

                                if(emailcount>0){
                                    ViewData["Message"]="The User Already Exists !!!!!";
                                }
                                else if(numcount>0){
                                    ViewData["Message"]="The User Already Exists !!!!!";
                                }
                                else{
                                            employee.EmployeeTechnology = "Not Assigned";
                                            employee.EmployeeProject = "Not Assigned";
                                            employee.EmployeeWorkingStatus = "Not Assigned" ;
                                            employee.EmployeeTrainingStartTime = "Not Assigned'
                                            employee.EmployeeTrainingEndTime = "Not Assigned"
                                            employee.EmployeeTrainerName = "Not Assigned" ;
                                            employee.EmployeeVacationStartTime = "Not Assigned'
                                            employee.EmployeeVacationEndTime = "Not Assigned"
```

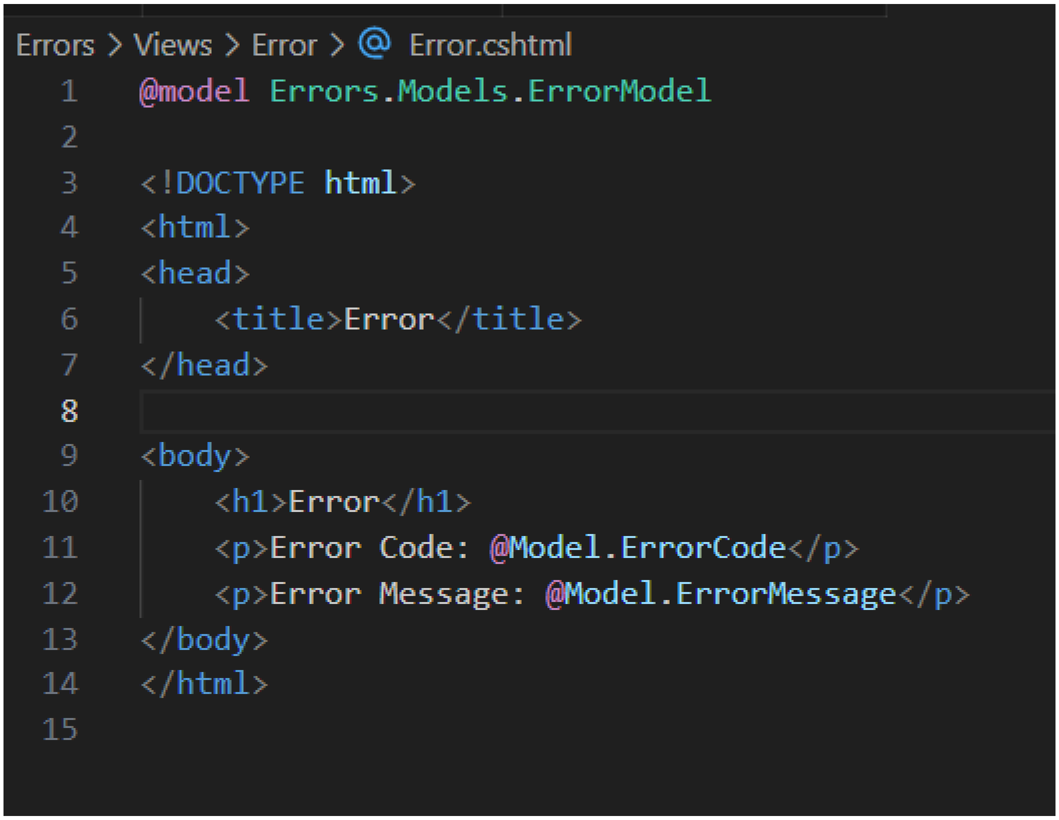


```
                                            employee.EmployeeTrainerName = "Not Assigned" ;
                                            employee.EmployeeVacationStartTime = "Not Assigned"
                                            employee.EmployeeVacationEndTime = "Not Assigned" ;
                                            employee.EmployeeVacationStatus =  "Not Assigned" ;
                                            employee.EmployeeImage = null;
                                            employee.EmployeeVacationReason = "Not Assigned";
                                    ViewData["Message"]="Succesfully Registered , Log in to Worl
                                    _dbContext.Add(employee);
                                    await _dbContext.SaveChangesAsync();
                                }
                            }

                }
            }
            catch(Exception exception){
                ModelState.AddModelError(String.Empty,$"Something Went Wrong{exception.Message}");
            }
        }
        ModelState.AddModelError(String.Empty,$"Something Went Wrong Invalid Model");
        return View(employee);
    }
    [HttpGet]

    public IActionResult forgotPasswordE()
    {
        return View();
    }
```

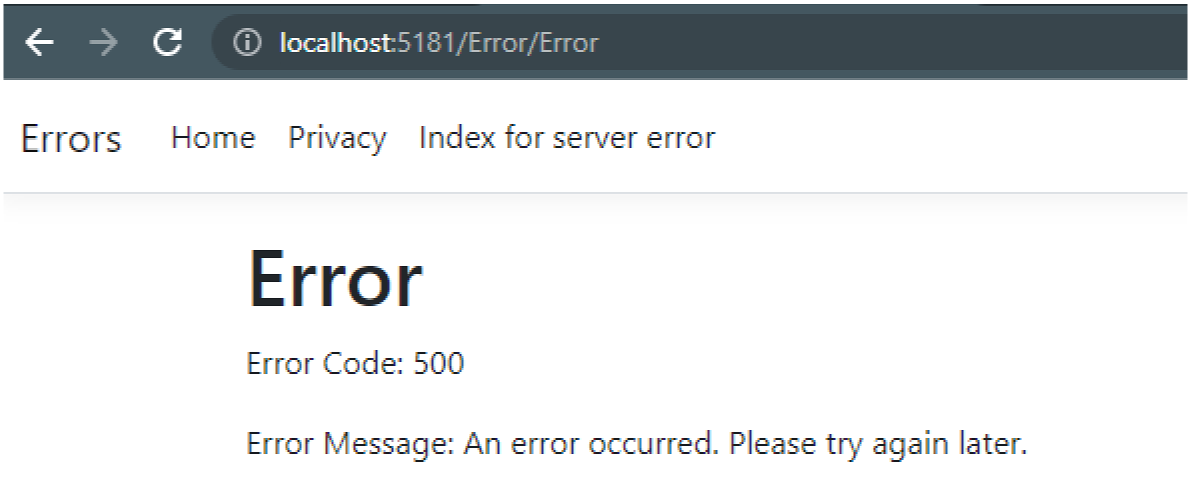
Server Error:



```csharp
using Microsoft.AspNetCore.Mvc;
using Errors.Models;

namespace Errors.Controllers
{
    public class ErrorController : Controller
    {
        public IActionResult Error()
        {
            var errorModel = new ErrorModel
            {
                ErrorCode = 500,
                ErrorMessage = "An error occurred. Please try again later."
            };
            return View(errorModel);
        }
    }
}
```



```html
@model Errors.Models.ErrorModel

<!DOCTYPE html>
<html>
<head>
    <title>Error</title>
</head>

<body>
    <h1>Error</h1>
    <p>Error Code: @Model.ErrorCode</p>
    <p>Error Message: @Model.ErrorMessage</p>
</body>
</html>

```



Error

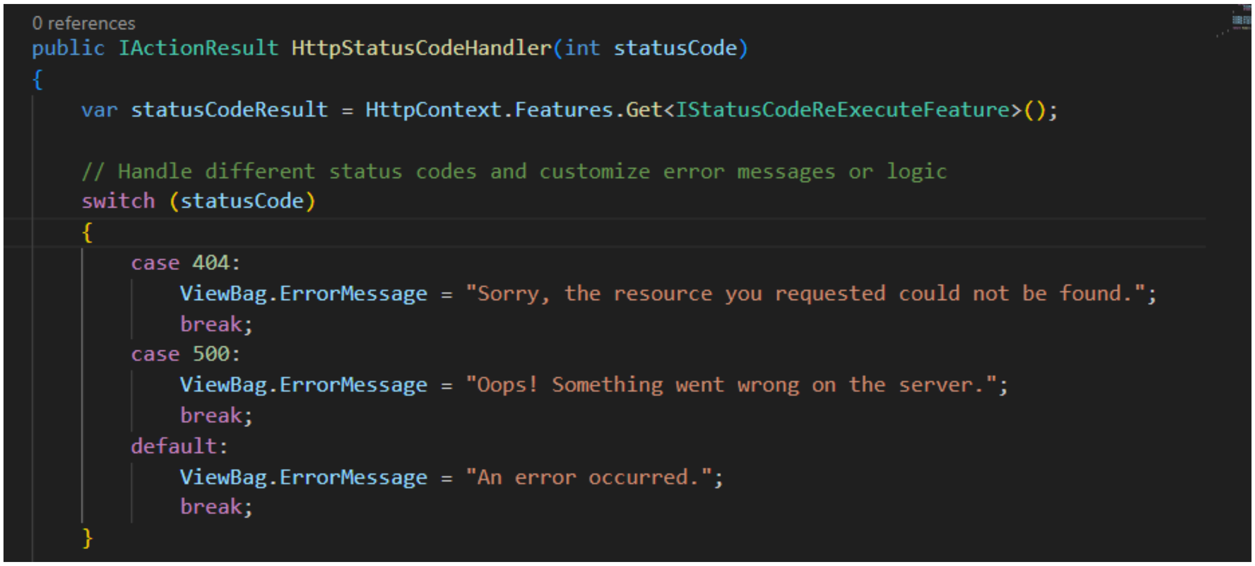
Error Code: 500

Error Message: An error occurred. Please try again later.

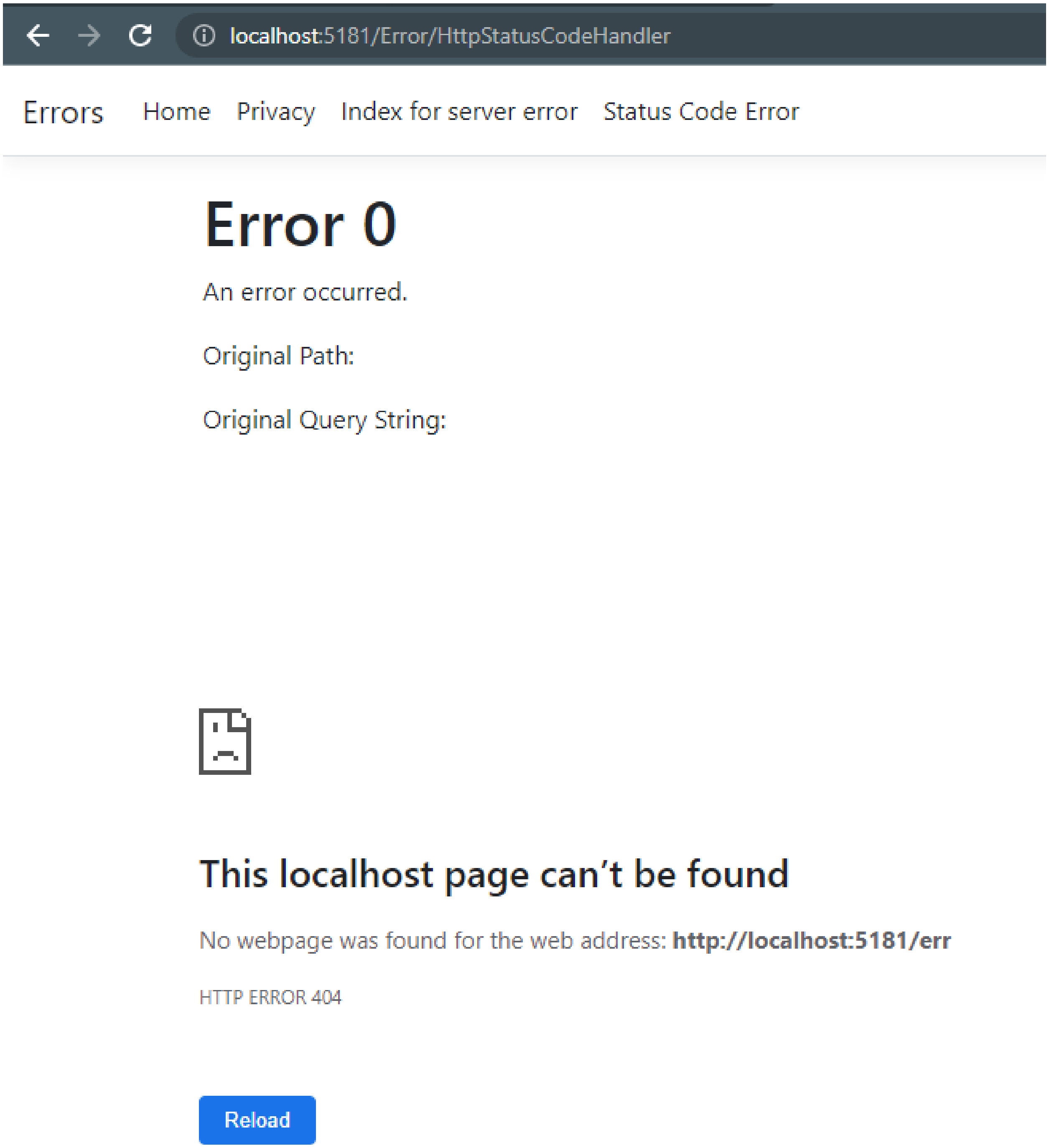
Database Error:

If a database error occurs (such as a unique constraint violation or a connection issue), it is database error.

Status Code Error:



```
public IActionResult HttpStatusCodeHandler(int statusCode)
{
    var statusCodeResult = HttpContext.Features.Get<IStatusCodeReExecuteFeature>();

    // Handle different status codes and customize error messages or logic
    switch (statusCode)
    {
        case 404:
            ViewBag.ErrorMessage = "Sorry, the resource you requested could not be found.";
            break;
        case 500:
            ViewBag.ErrorMessage = "Oops! Something went wrong on the server.";
            break;
        default:
            ViewBag.ErrorMessage = "An error occurred.";
            break;
    }
```



Error 0

An error occurred.

Original Path:

Original Query String:

This localhost page can’t be found

No webpage was found for the web address: **http://localhost:5181/err**

HTTP ERROR 404