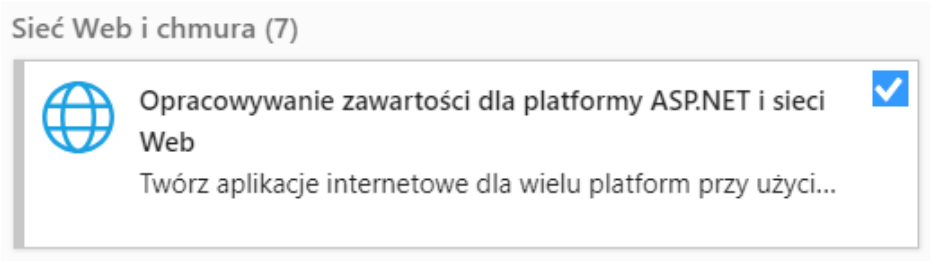
Advanced Programming C#

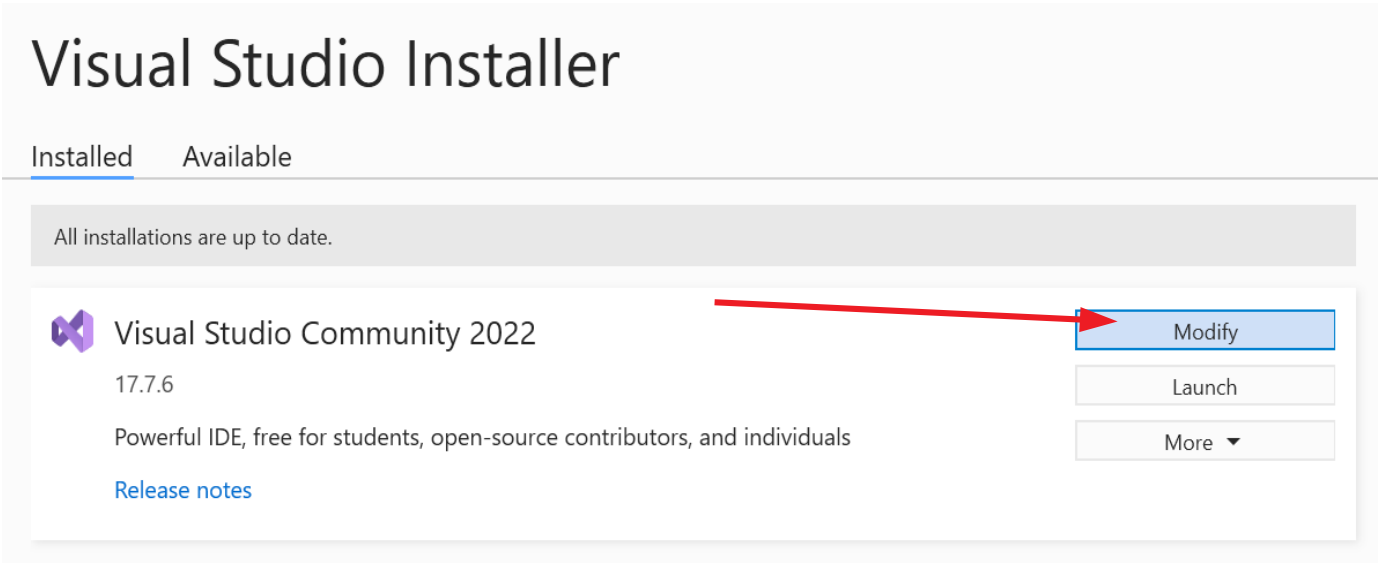
Lecture 12

dr inż. Małgorzata Janik

malgorzata.janik@pw.edu.pl

Today you will need:

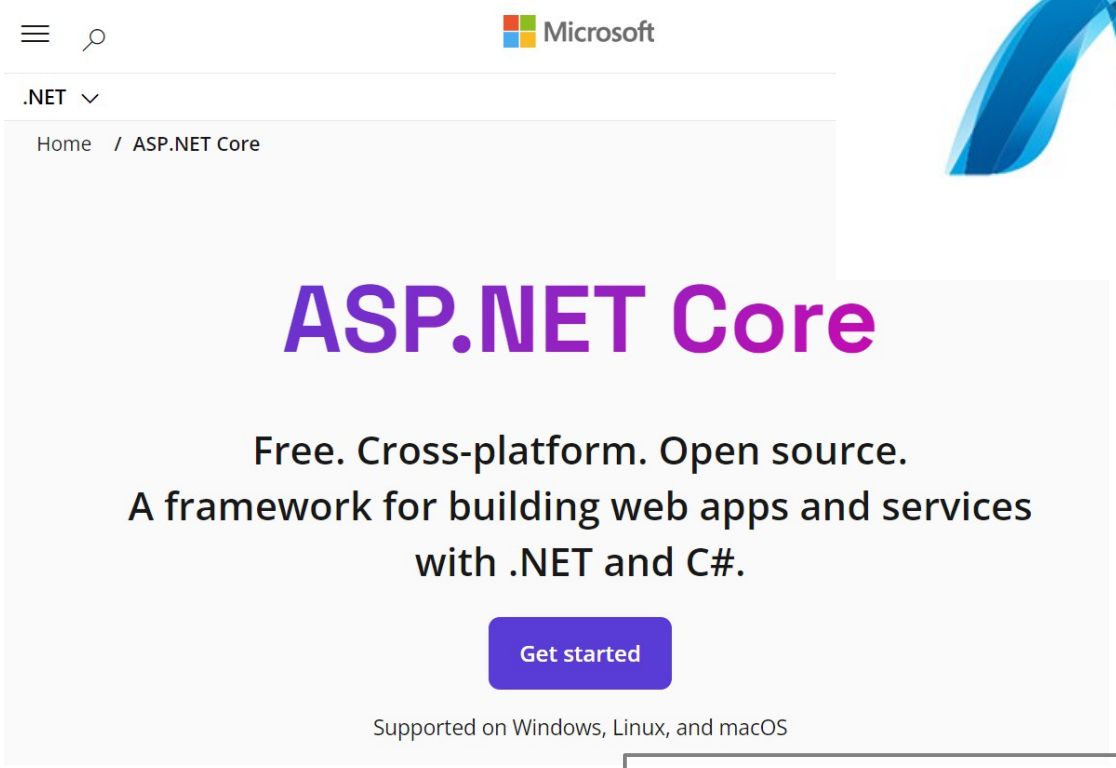




ASP .NET
(Web Forms)

ASP .NET

ASP stands for **A**ctive **S**erver **P**ages



Microsoft® .NET™

ASP.NET is an open source web framework for building modern web apps and services with .NET. ASP.NET creates websites based that are simple, fast, and can scale to millions of users.

https://www.asp.net/

Tasks

Demonstration

- Intro: create new ASP Web App Project, check it out, make some simple modifications

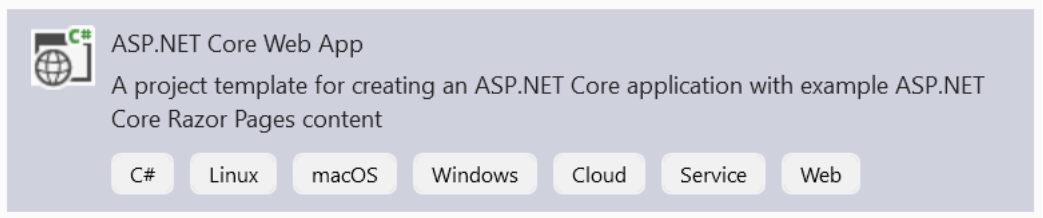
- File upload: allow users to upload their pictures into appropriate directory on the server
- Gallery: display uploaded pictures side by side
- If picture is clicked, bigger version should be displayed below

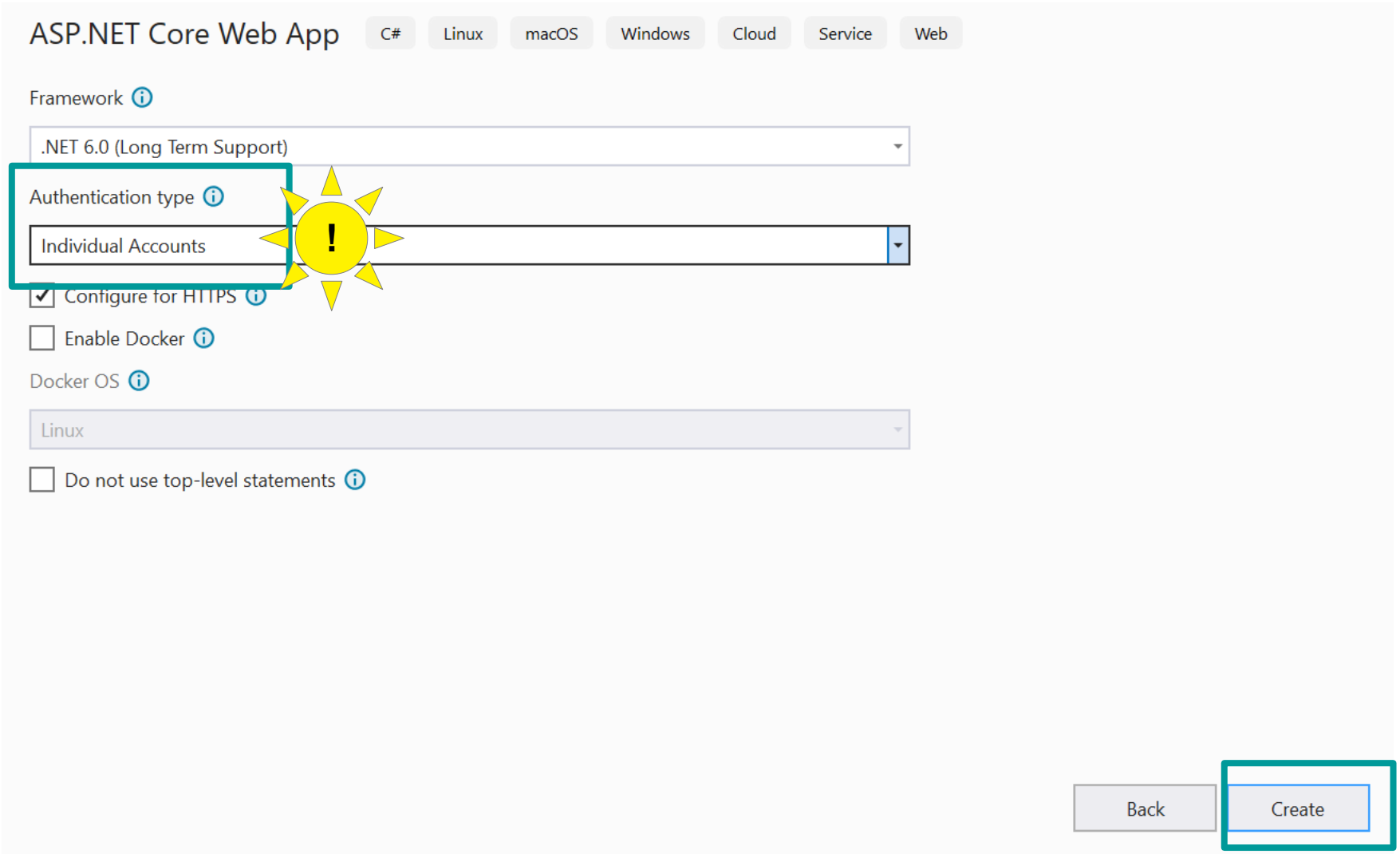
Tasks: Intro

- Create new ASP.NET Core Web App: „PhotoGallery”
  - Add user authentication
- Run the project in the browser
  - Try to register & login, remember to use password with special character!
- Modify the „Home” web page
- Add new Page → „Gallery”. Add it to the main menu (at the top)
  - Project → Pages → Add... → Razor Page...
- Prepare Photo Gallery service

ASP .NET Web Forms

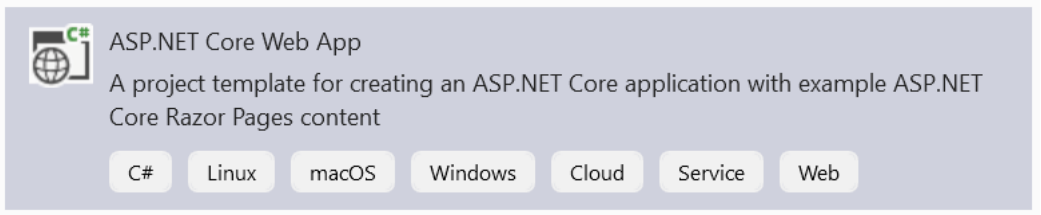
- New Project
- ASP.NET Core Web App
- Authentication





ASP .NET Web Forms

- New Project
- ASP.NET Core Web App
- Authentication
  - To finalize authentication database must be migrated:



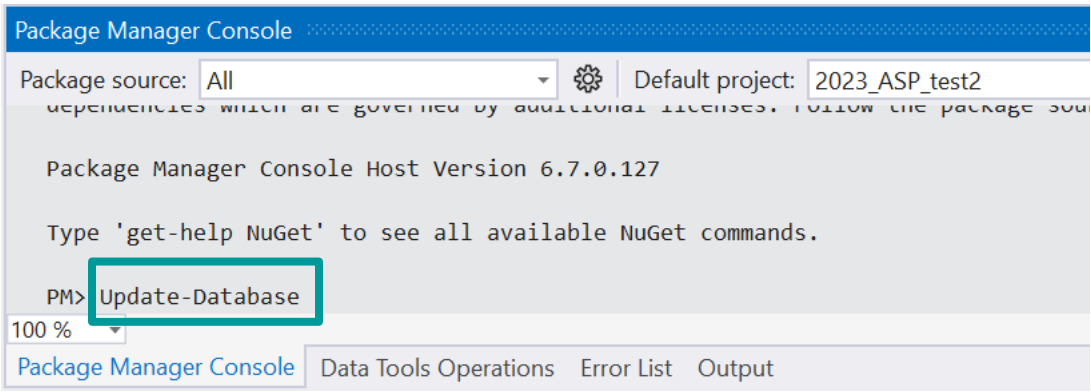
Apply the migrations to initialize the database.

Visual Studio | .NET Core CLI

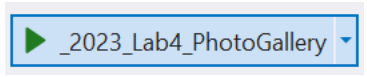
Run the following command in the Package Manager Console (PMC):

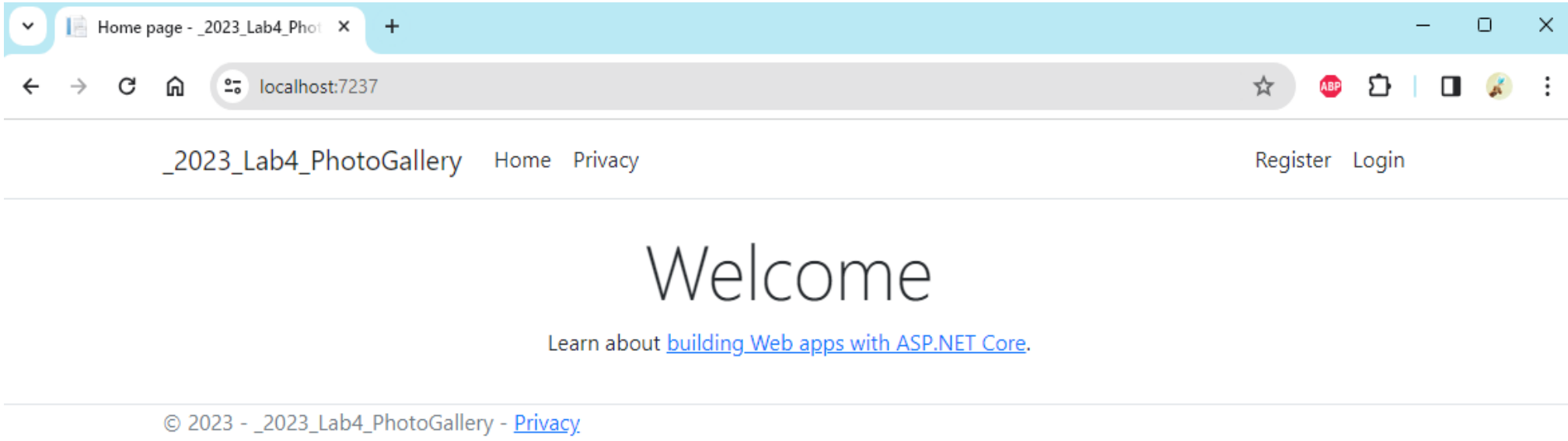
```
Update-Database
```

- Tools → NuGet Package Manager → Package Manager Console



Web Forms: run default project





- Try to register & login, remember to use password with special character!

  (click the link to confirm the e-mail, the full procedure is not implemented)

Modification of „Home” webpage

- modify Index.cshtml

```
@page
@model IndexModel
@{
    ViewData["Title"] = "Home page";
}

<div class="text-center">
    <h1 class="display-4">Welcome to Your Photo Gallery!</h1>

    <h3>Our platform allows to create photo gallery. You can browse photos any time
                                                        from any device.</h3>
    <p>Register, log in, add new photos and browse!</p>

</div>
```

Modifications of the master webpage

File: **_Layout.cshtml** (Pages → Shared)

Name of the application

```
<title>@ViewData["Title"] - Your Photo Gallery</title>
<a class="navbar-brand" asp-area="" asp-page="/Index">Your Photo Gallery</a>
```

Comment the code

```
<%--  -->
```

Adding new pages to the main menu (Layout)

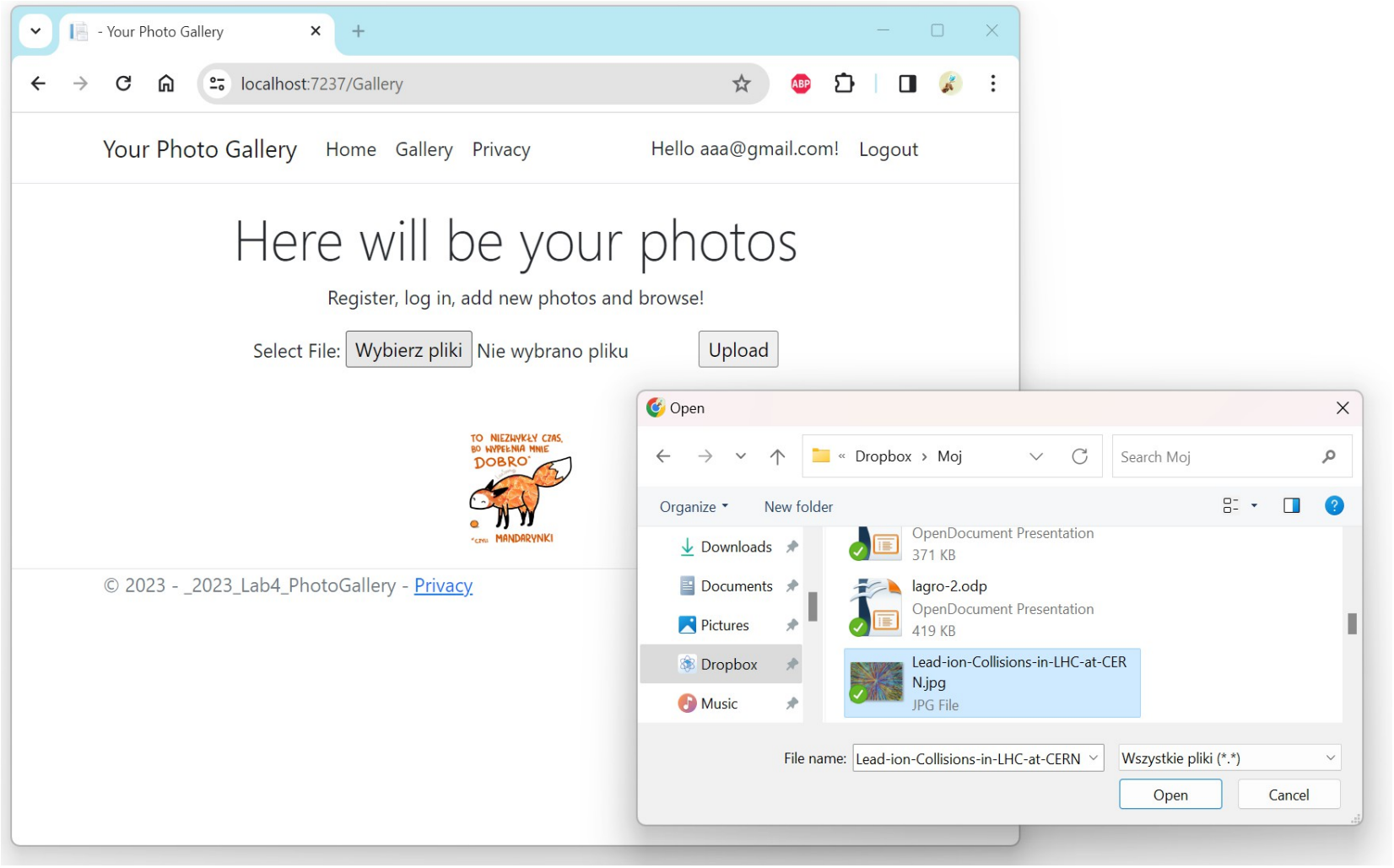
1) Project → Pages → Add... → Razor Page...
2) Code below should be modified:

Adding Gallery page!

```
<ul class="navbar-nav flex-grow-1">
        <li class="nav-item">
            <a class="nav-link text-dark" asp-area="" asp-page="/Index">Home</a>
        </li>
        <li class="nav-item">
            <a class="nav-link text-dark" asp-area="" asp-page="/Gallery">Gallery</a>
        </li>
    ...
```

Tasks: Photo Gallery – File upload

- Idea:

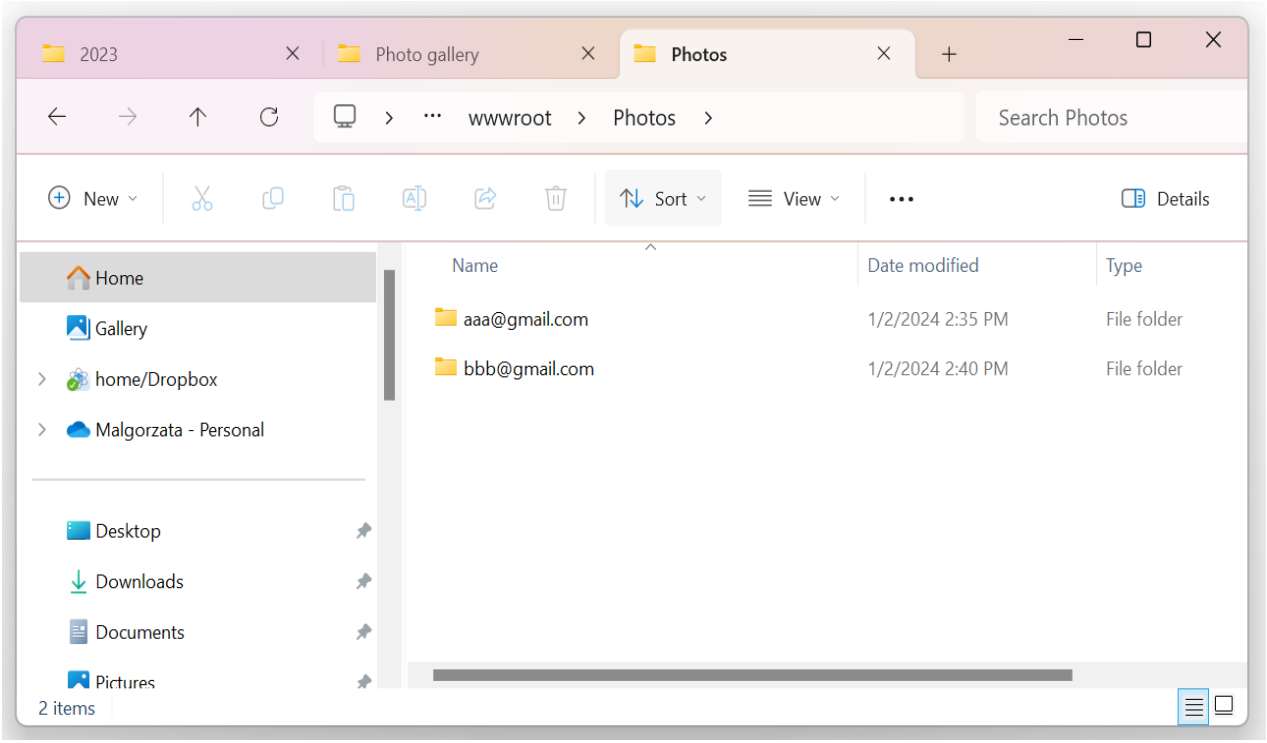


Tasks: Photo Gallery – File upload

- Uploaded files should be saved in „Photos” folder (located in the project wwwroot folder) in the user directory
  - ~/Photos folder should be created by hand (once)

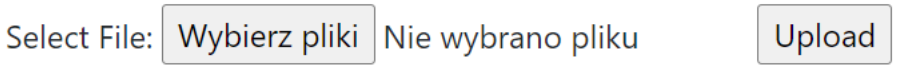
```
Path.Combine(this.Environment.WebRootPath, "Photos");
```

  - User folders should be created by the application
  - Photos should be stored in appropriate user folders



Tasks: Photo Gallery – File upload

- Add to the Gallery.cshtml file form for the upload:




```
<form method="post" enctype="multipart/form-data">
    <span>Select File:</span>
    <input type="file" name="postedFiles" multiple />
    <input type="submit" value="Upload" asp-page-handler="Upload" />
    <br/>
    <span style="color:green">@Html.Raw(Model.Message)</span>
</form>
```

Label

Select files

Button

Label with message

Gallery.cshtml.cs

```
public class GalleryModel : PageModel
{
    private Microsoft.AspNetCore.Hosting.IWebHostEnvironment Environment;
    public string Message { get; set; } // message to be displayed to user
    public IEnumerable<string> Images { get; set; } = new List<string>(); //paths to the images to be shown


        public GalleryModel(IWebHostEnvironment _environment)
        {
            Environment = _environment;


        }

       public void OnPostUpload(List<IFormFile> postedFiles) //method called when button „Upload” is clicked
       {
             //check if user is logged in
             //create a path file to save photos
             //make foreach loop over postedFiles
             //check if each file is ok
             //save files to the server
       }


        public void OnGet()
        {
             //place to populate Images list with relative paths to all uploaded files
         }

}
```

Tasks: Photo Gallery – File upload

- Inside OnPostUpload method
  - Check:
    - If folder exists
      - Get folder path+name:

```
this.Environment.WebRootPath ← Project wwwroot folder

User.Identity.Name ← user login; so:

string path = Path.Combine(this.Environment.WebRootPath, "Photos");

string photoFolder = Path.Combine(path, User.Identity.Name);
```

      - Check if folder exists:

```
Directory.Exists(photoFolder)
```

      - If not, create folder:

```
Directory.CreateDirectory(photoFolder);
```

Tasks: Photo Gallery – File upload

- Inside OnPostUpload method

Check inside `foreach (IFormFile postedFile in postedFiles)` loop:

- If file type is one of the following: jpg / gif / png / bmp?

```
(postedFile.ContentType == "image/jpeg") ||
(postedFile.ContentType == "image/png") || ...
```

  - If not, write appropriate message, e.g. :

```
this.Message += "Wrong type of file";
```

- If file size is < 10 MB?

```
Convert.ToInt64(postedFile.ContentLength)
```

  - If not, write appropriate message

Extra

Tasks: Photo Gallery – File upload

- Inside OnPostUpload method, in the foreach loop
  - Create new name for the uploaded file:
    - We want the name to be unique (if client uploades two files with the same name, we want to be able to store both)
    - We have to change default file name: insert timestamp
      - NameOfFile.jpg → NameOfFile.636138997008655118.jpg

Comes from timestamp

    - Useful methods:

`Path.GetFileNameWithoutExtension(FileUpload.FileName)` → returns filename only, without extension

`DateTime.Now.Ticks.ToString()` → returns series of numbers based on the current time

`Path.GetExtension(FileUpload.FileName)` → returns extension (string)

`Path.Combine(photoFolder, name + time + extension)` → „Path.Combine()" method allows to create strings representing properly defined paths

Tasks: Photo Gallery – File upload

- Inside OnPostUpload method
  - Upload the file

```
using (FileStream stream = new FileStream(Path.Combine(photoFolder, newFileName), FileMode.Create))
{
    postedFile.CopyTo(stream);
    uploadedFiles.Add(newFileName);
}
```

  - Change status label

```
this.Message += string.Format("<b>{0}</b> uploaded.<br />", postedFile.FileName);
```

  - Try it!

Tasks: Photo Gallery – Display Photos

- Change Gallery.cshtml file: add images in a loop
  - With „img” as the small 100 x 100 px icon:

```
@foreach (var image in Model.Images)
 {
    <img src="@Url.Content(image)" alt="@image" width="100" height="100" />
 }
```

- Change Gallery.cshtml.cs file:
  - Work on the new function (see next slide)

```
public void OnGet()
        { }
```

Tasks: Photo Gallery – Display Photos

- `OnGet` function

This function should fill the list `IEnumerable<string> Images` with relative paths to all uploaded photos of the current user

- Get paths of all photos that should be displayed
  - Get filenames of photos

```
string[] allPhotoFiles = new DirectoryInfo(photoFolder).GetFiles().Select(o => o.Name).ToArray();
```

  - Create list of filenames with „Photos/login” included

```
Create List/Array<String> where the proper paths will be stored.
```

Use a loop to iterate over `allPhotoFiles` array. For each string stored in array (for each file) add directories to the created list of strings. Full path should consist of `"Photos/" + User.Identity.Name + "/" + Path.GetFileName(file).`

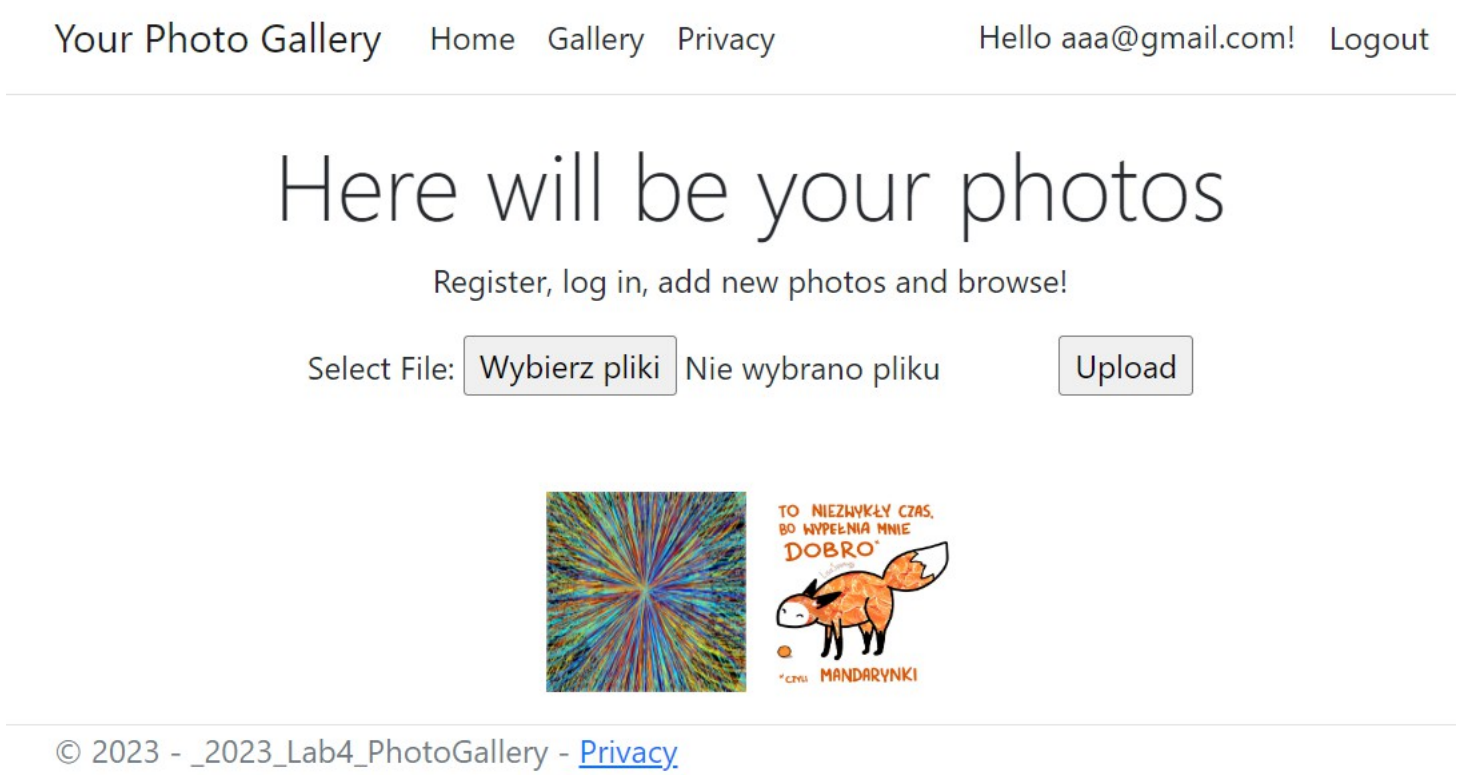
- Remember to set Image field to newly created list

Tasks: Photo Gallery – Display Photos

- After uploading a photo run OnGet() function

```
OnGet();
```

- Try it!



References & additional info

- Upload Photos to your website – YouTube tutorial

  https://www.youtube.com/watch?v=dPUmQBowLfs

- ASP .NET

  https://www.asp.net/

KONIEC

dr inż. Małgorzata Janik
malgorzata.janik@pw.edu.pl

Tasks: Photo Gallery

- After clicking a photo we would like it to be enlarged.
  - Add an image control below repeater that will hold enlarged image
    - `<img id="preview" src="" alt="" width="400" />`
  - Add attribute to the img controls inside reapeter:
    - `onmouseover="preview.src=this.src"`