



Advanced Programming C#

Lecture 4

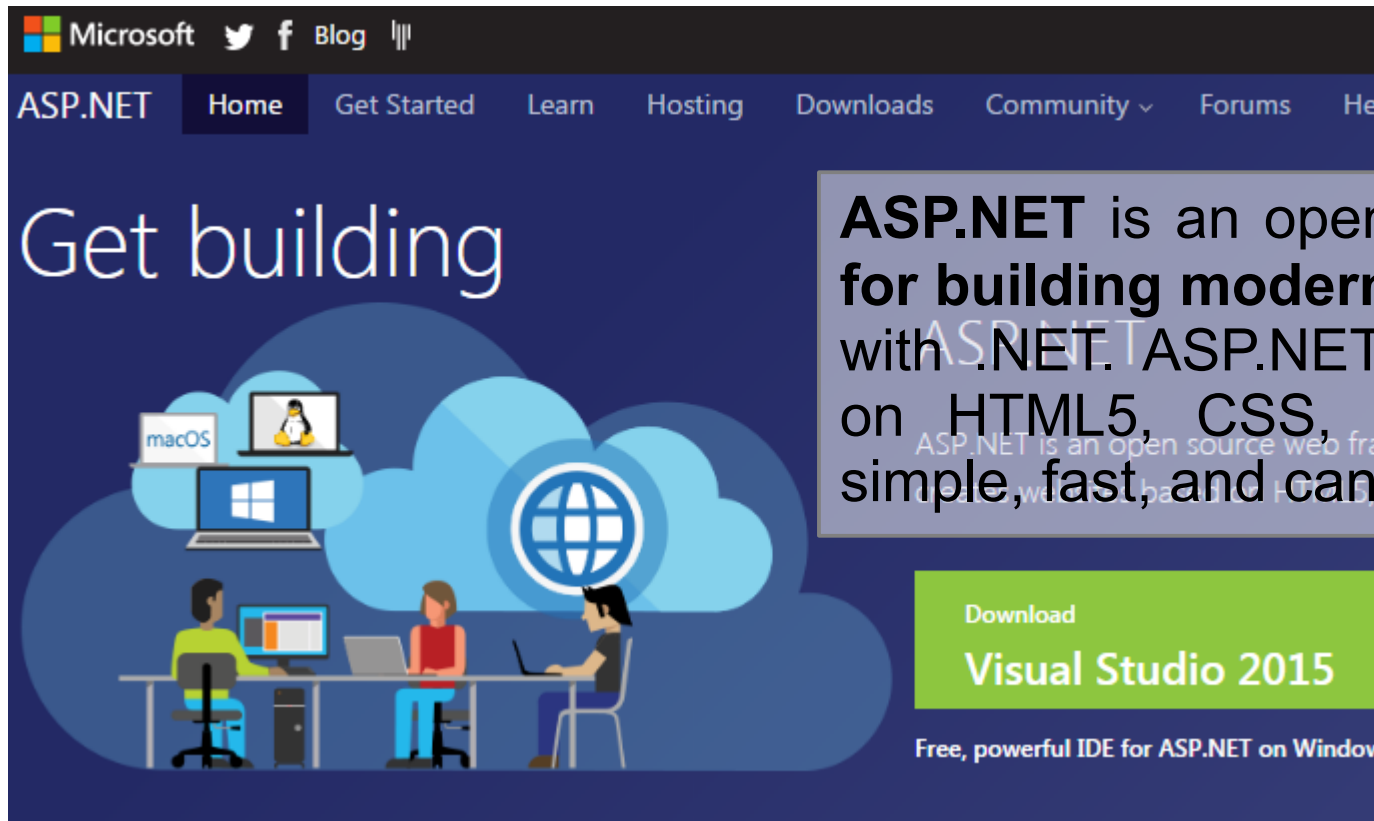
dr inż. Małgorzata Janik
majanik@if.pw.edu.pl

Winter Semester 2017/2018



ASP .NET (Web Forms)

ASP.NET



ASP.NET is an open source **web framework for building modern web apps and services** with .NET. ASP.NET creates websites based on HTML5, CSS, and JavaScript that are simple, fast, and can scale to millions of users.

<https://www.asp.net/>

ASP .NET

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



ASP is a development framework for building web pages.

ASP supports several different development models:

ASP.NET Web Forms

ASP.NET MVC

ASP.NET Web Pages

ASP.NET Single Page Application

ASP .NET Web Forms

ASP.NET Web Forms is a part of the ASP.NET web application framework and is included in Visual Studio.

ASP.NET Web Forms are:

- Based on Microsoft ASP.NET technology, in which **code that runs on the server dynamically generates Web page output to the browser or client device**.
- Compatible with any browser or mobile device. An ASP.NET Web page automatically renders the correct browser-compliant HTML for features such as styles, layout, and so on.
- Compatible with any language supported by the .NET common language runtime, such as Microsoft Visual Basic and Microsoft Visual C#.
- Flexible because you can add user-created and third party controls to them.

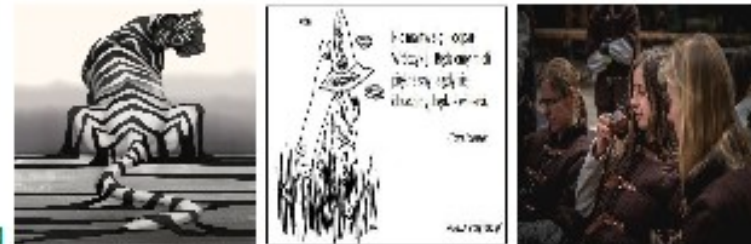
ASP.NET Web Forms offer:

- **Separation of HTML and other UI code** from application logic.
- **A rich suite of server controls for common tasks**, including data access.
- Powerful data binding, with great tool support.
- Support for client-side scripting that executes in the browser.
- Support for a variety of other capabilities, including routing, security, performance, internationalization, testing, debugging, error handling and state management.

<https://www.asp.net/web-forms/what-is-web-forms>

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

Photo Gallery

 Successfully uploaded file witcher1.jpg

- File upload: allow users to upload their pictures into appropriate directory on the server
- Gallery: display uploaded pictures side by side
- Javascript (full screen): if picture is clicked, it should be displayed in the full screen mode

Tasks: Intro

- Create new Web Forms Project: „PhotoGallery”
- Run the project in the browser
 - Try to register & login, remember to use password with special character!
- Modify the „About” web page
- Add new Web Form (with Master Page) → „Gallery”. Add it to the main menu (at the top)
 - Project → Add new... → Web Form with Master Page
- Prepare Photo Gallery service

ASP .NET Web Forms

- New Project

The screenshot illustrates the process of creating a new ASP.NET Web Forms project in Visual Studio. It shows three overlapping windows:

- New Project:** The 'Visual C#' category is expanded, and 'ASP.NET Web Application (.NET Framework)' is selected. The framework is set to '.NET Framework 4.6.1'.
- New ASP.NET Web Application - Test_Login_WebApplication1:** The 'Web Forms' template is selected. The description states: 'A project template for creating ASP.NET Web Forms applications. ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model. A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with data access.' The 'Change Authentication' button is visible.
- Change Authentication:** The 'Individual User Accounts' option is selected. The description states: 'For applications that store user profiles in a SQL Server database. Users can register, or sign in using their existing account for Facebook, Twitter, Google, Microsoft, or another provider.'

At the bottom of the 'New ASP.NET Web Application' window, the 'Add folders and core references for:' section shows 'Web Forms' checked, and 'Add unit tests' is unchecked. The 'Test project name' is 'Test_Login_WebApplication1.Tests'.

Web Forms: run default project

Application name Home About Contact

Register Log in

ASP.NET

ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS, and JavaScript.

[Learn more »](#)

Getting started

ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model. A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with data access.

[Learn more »](#)

Get more libraries

NuGet is a free Visual Studio extension that makes it easy to add, remove, and update libraries and tools in Visual Studio projects.

[Learn more »](#)

Web Hosting

You can easily find a web hosting company that offers the right mix of features and price for your applications.

[Learn more »](#)

© 2016 - My ASP.NET Application

Modification of „About” webpage

```
<asp:Content ID="BodyContent"
ContentPlaceHolderID="MainContent" runat="server">
    <h2>Your Photo Gallery </h2>
    <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>
</asp:Content>
```

Modifications of the master webpage

File: **Site.master**

Name of the application

```
<a class="navbar-brand" runat="server"
href="~/ ">Your Photo Gallery</a>
```

Comment the code

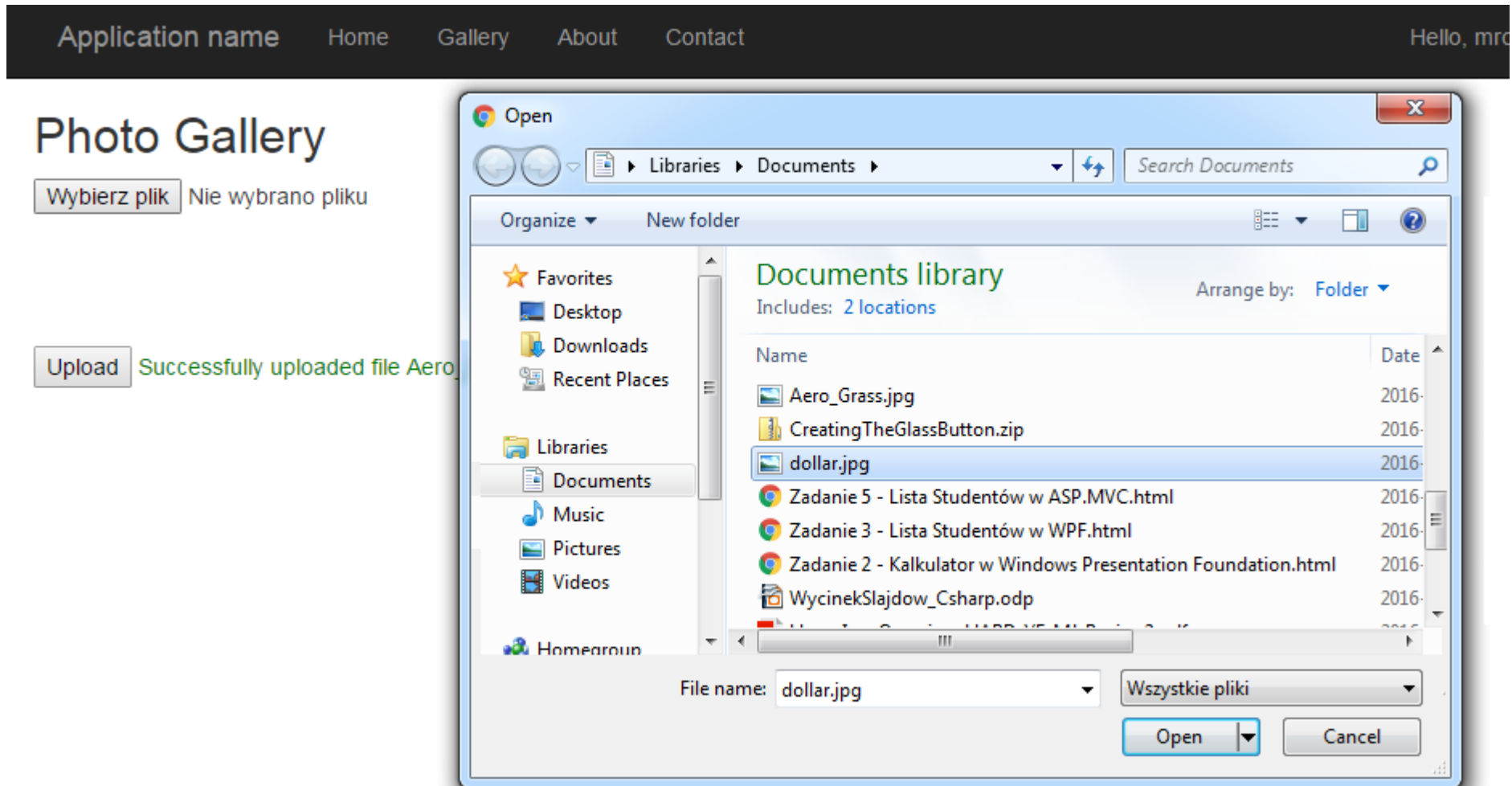
```
<%-- -->
```

Adding new pages to the main menu (site.master)

```
<div class="navbar-collapse collapse">
  <ul class="nav navbar-nav">
    <li><a runat="server" href="~/ ">Home</a></li>
    <li><a runat="server" href="~/About">About</a></li>
    ...
```

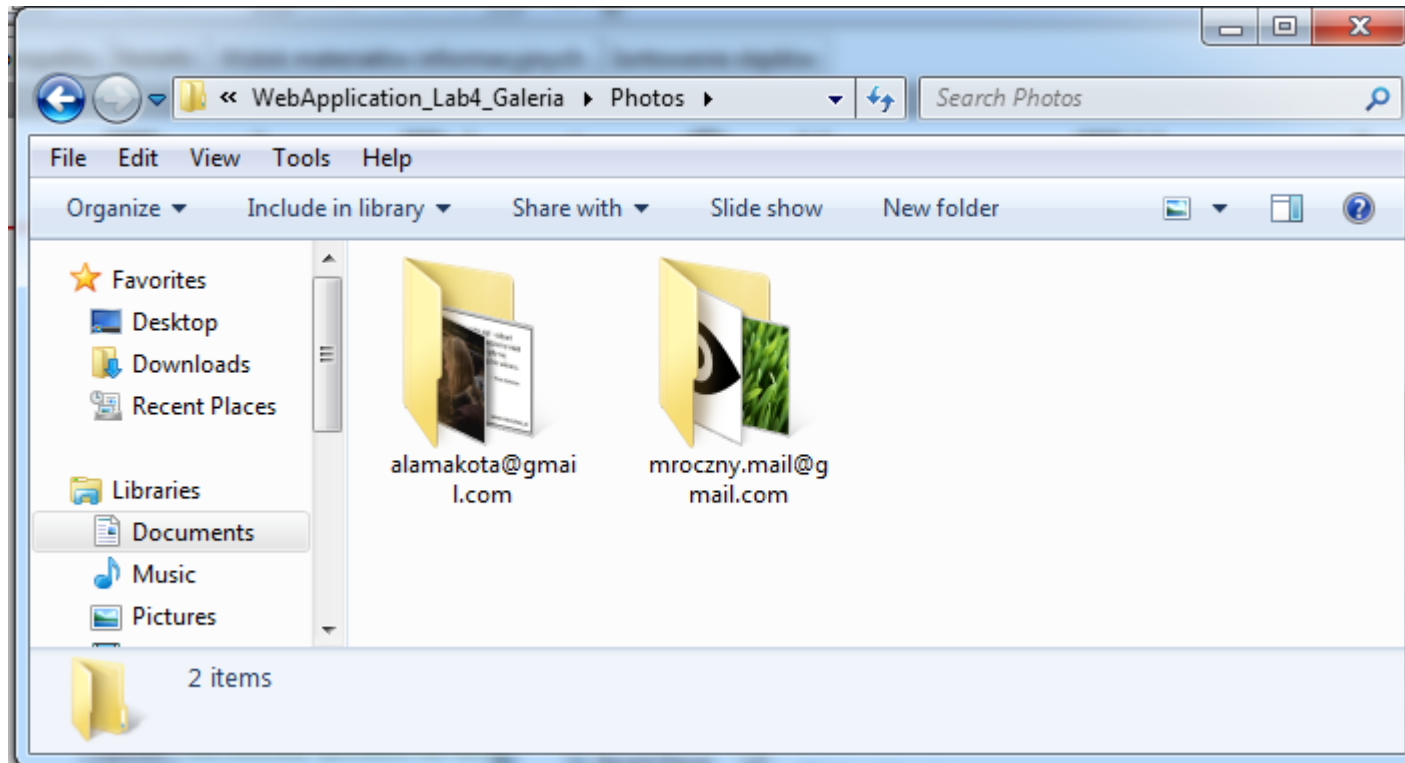
Tasks: Photo Gallery – File upload

- Idea:



Tasks: Photo Gallery – File upload

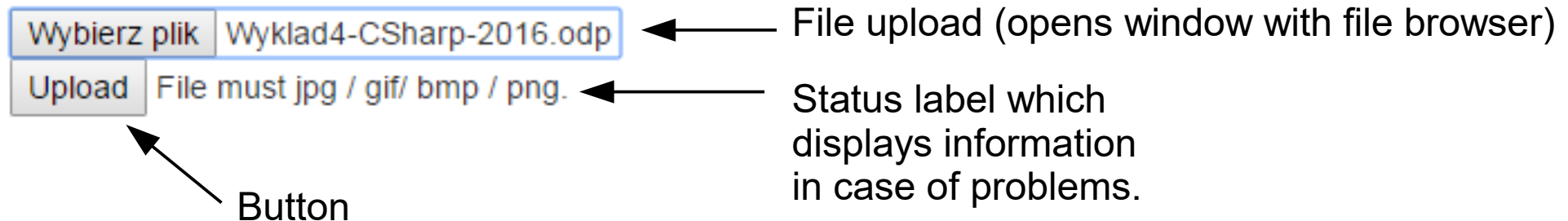
- Uploaded files should be saved in „Photos” folder (located in the project home folder) in the user directory
 - ~/Photos folder should be created by hand (once)
 - User folders should be created by the application
 - Photos should be stored in appropriate user folders



Tasks: Photo Gallery – File upload

- Idea:

Photo Gallery



- Add controls:
 - **FileUpload** control
 - Upload **button**
 - **Label** for the status display

Tasks: Photo Gallery – File upload

- Create OnClick event for upload button
 - Check:
 - If FileUpload has a file uploaded?
 - FileUpload.HasFile
 - If not, write appropriate text in status label
 - If file type is one of the following: jpg / gif / png / bmp?
 - (FileUpload.PostedFile.ContentType == "image/jpeg") ||
 - (FileUpload.PostedFile.ContentType == "image/png") || ...
 - If not, write appropriate text in status label
 - If file size is < 10 MB?
 - Convert.ToInt64(FileUpload.PostedFile.ContentLength)
 - If not, write appropriate text in status label

Tasks: Photo Gallery – File upload

- Inside OnClick event for upload button

- Check:

- If folder exists

- Get folder path+name:

```
string photoFolder =  
    Path.Combine(Server.MapPath("~/Photos/"),  
        User.Identity.Name);
```

- Check if folder exists:

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

- If not, create folder:

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

Tasks: Photo Gallery – File upload

- Inside OnClick event for upload button
 - Create new name for the uploaded file:
 - We want the name to be unique (if client uploads 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
 - Useful methods:

Comes from timestamp

`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 OnClick event for upload button

- Upload the file

```
FileUpload.SaveAs(full_file_path);
```

- Change status label

```
LabelStatus.Text = "<font color='green'> Successfully  
uploaded file " + FileUpload.FileName + "</font>";
```

- Try it!

Tasks: Photo Gallery – Display Photos

- Change aspx file: add Repeater control
 - With „img” as the repeated item template

```
<asp:Repeater ID="RepeaterUserPhoto" runat="server">  
    <ItemTemplate>  
  
  
  
    </ItemTemplate>  
  
</asp:Repeater>
```

- Change aspx.cs file:
 - Add new function

```
public void DisplayUploadedPhotos(string folder)  
{ }
```

Tasks: Photo Gallery – Display Photos

- DisplayUploadedPhotos function
 - As the name suggest, should display all uploaded photos of the current user
 - Get paths of all photos that should be displayed
 - Get filenames of photos

```
string[] allPhotoFiles = Directory.GetFiles(folder);
```

- Create list of filenames with full path included

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

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

- Connect list as a source for the repeater.

```
RepeaterUserPhoto.DataSource = pathList;
```

```
RepeaterUserPhoto.DataBind();
```

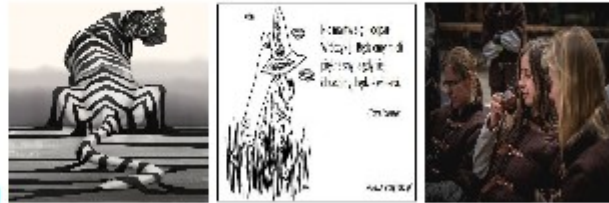
Tasks: Photo Gallery – Display Photos

- After uploading a photo run DisplayUploadedPhotos function

```
DisplayUploadedPhotos(photoFolder);
```

- Try it!

Photo Gallery

 Successfully uploaded file witcher1.jpg

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

- ``

- Add attribute to the img controls inside reapeater:

- `onmouseover="preview.src=this.src"`

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
majanik@if.pw.edu.pl

Winter Semester 2016/2017