**WebForms**

* IIS – IIS Express is installed default, but if we want to use IIS, we need to install it and make some configurations
* Project structure:
  + App\_Data folder – where we place the local database files. The only place where our application has write rights
  + App\_Start folder –
  + Content folder – there are all the CSS files
  + fonts folder – if we need any custom fonts for the app here can be placed
  + Scrips – contains the Javascript files
  + Web.config - responsible for configuring the application (connection string, application settings, …)
  + Global.asax - very important file! It’s about the general look and feel of the application, contains code, that will affect the whole application
* how the content gets created:
  + Site.Master - it’s a layout page, has a content placeholder in it (this will be replaced with the content – from .aspx files)
* .aspx files:
  + it starts with a page directive and it has properties (Title, Language, MasterPageFile, CodeBehind, …)
  + the properties values will be replaced inside the variables on the page
  + when .NET sees an .aspx page it will automatically generate a C# file
  + : does the same thing that = after <% code directive, but it does not encoding (not transfers HTML tags, writes out as they are )
  + common synthax elements:
    - <%@ %> - Page directives
    - <asp:SomeControl … ruant=”server”> … </asp:SomeControl> - Controls
    - <% %> - Inline code (no good practice to include much code inside HTML)
    - <%= %> - Write to the response stream (can put a variable inside it)
    - <%: %> - Same as above, but encoded
    - <%$ %> - Expression evaluation
    - <%# %> - Data binding
    - <%# %> - Encoded data binding
    - <%-- --%> - Comments
* code behind files:
  + .aspx.cs
  + if not necessary we don’t want to add code to the .aspx file (don’t want sphagetti code)
  + good to place code here, in the code behind file
  + every time when we adding something to .aspx page (control, button, …) the interaction needs to be happen here in the code behind file (.aspx.cs)
  + .aspx.designer.cs file - it’s also a partial class as the .aspx.cs file. (they are the same)
  + when adding a Control to the. aspx page, the Control definition will end up in the .aspx.designer.cs file, while the interaction will happen in the .aspx.cs file.
* Web Controls:
  + it comes with a lot of predefined Controls
  + can choose from the Toolbox and drag it to the page (either in code, or in the design view)
  + ex.: can drag and drop a button to the page
    - right click: Properties
      * Text, ID, Event (Click), ….
    - it also generates automatically in the designer page
* Handling Postback Data:
  + IsPostBack is an important property (we are sending data from the client to the server)
* Page Lifecycle:
  + 3 distinct phases: Init, Load, Render
* Trace : if I specify this in the Page directive, the application will show automatically all the events that are happening while running
* hidden page can be accessed (trace.axd)
* Create Web Controls:
  + right click on the project – add new item – Web Forms User Control
  + .ascx file
  + new Control needs to be registered first
  + need to add public properties to the control to communicate with it
* Managing State:
  + ViewState contains the information