Module 3 Day 3

Web Applications

What makes an application?

- Program Data
  - ✓ Variables & .NET Data Types
  - ✓ Arrays
  - ✓ More Collections (list, dictionary, stack, queue)
  - ✓ Classes and objects (OOP)

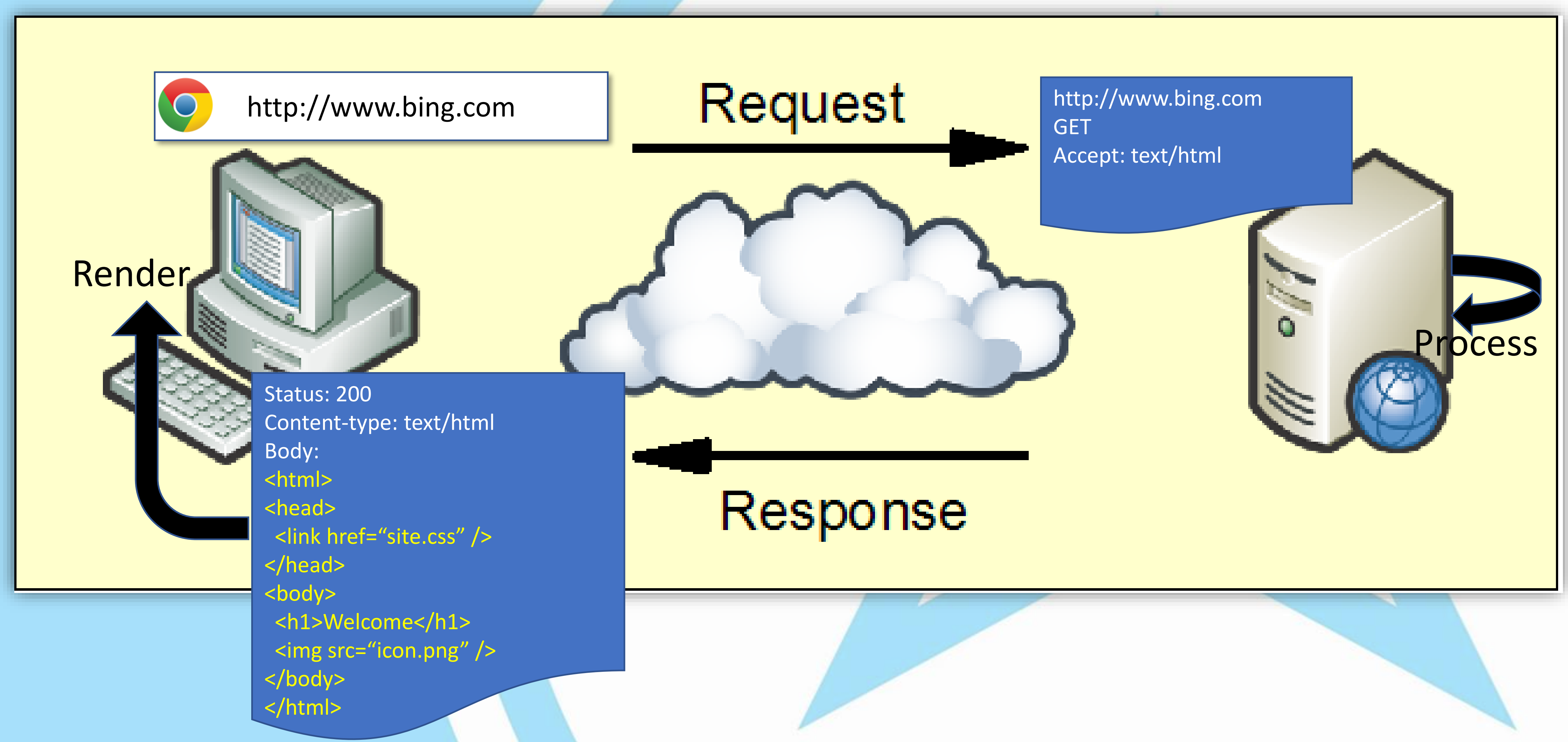
- Program Logic
  - ✓ Statements and expressions
  - ✓ Conditional logic (if)
  - ✓ Repeating logic (for, foreach, do, while)
  - ✓ Methods (functions / procedures)
  - ✓ Classes and objects (OOP)
  - ❖ Frameworks (MVC)

- Input / Output
  - User
    - ✓ Console read / write
    - ✓ HTML / CSS
    - ❑ Front-end frameworks (HTML / CSS / JavaScript)
  - Storage
    - ✓ File I/O
    - ✓ Relational database
    - ❑ APIs

HTTP Request-Response

- Request
  - URL – Uniform Resource Locator
  - HTTP Method / Verb (GET, PUT, POST, DELETE)
  - Headers (auth, content-type, cookies)
  - Body (sometimes)
- Response
  - Status (2xx, 3xx, 4xx, 5xx)
  - Headers (content-type, set-cookie)
  - Body (sometimes)

HTTP Request-Response



http://www.bing.com

Request

```
http://www.bing.com
GET
Accept: text/html
```

Process

Response

```
Status: 200
Content-type: text/html
Body:
<html>
<head>
  <link href=“site.css” />
</head>
<body>
  <h1>Welcome</h1>
  <img src=“icon.png” />
</body>
</html>
```

Render

HTTP Request-Response

- Stateless
  - server “remembers” nothing about the client between requests
- Cyclic
  - Response from server contains references, links and redirects
  - Client (e.g., Browser) responsible for making multiple requests to completely fulfill the user’s query and display a complete page
- Browser Developer Tools (F12)
  - Help you see all the requests that are taking place
  - Help understand performance issues

Demo

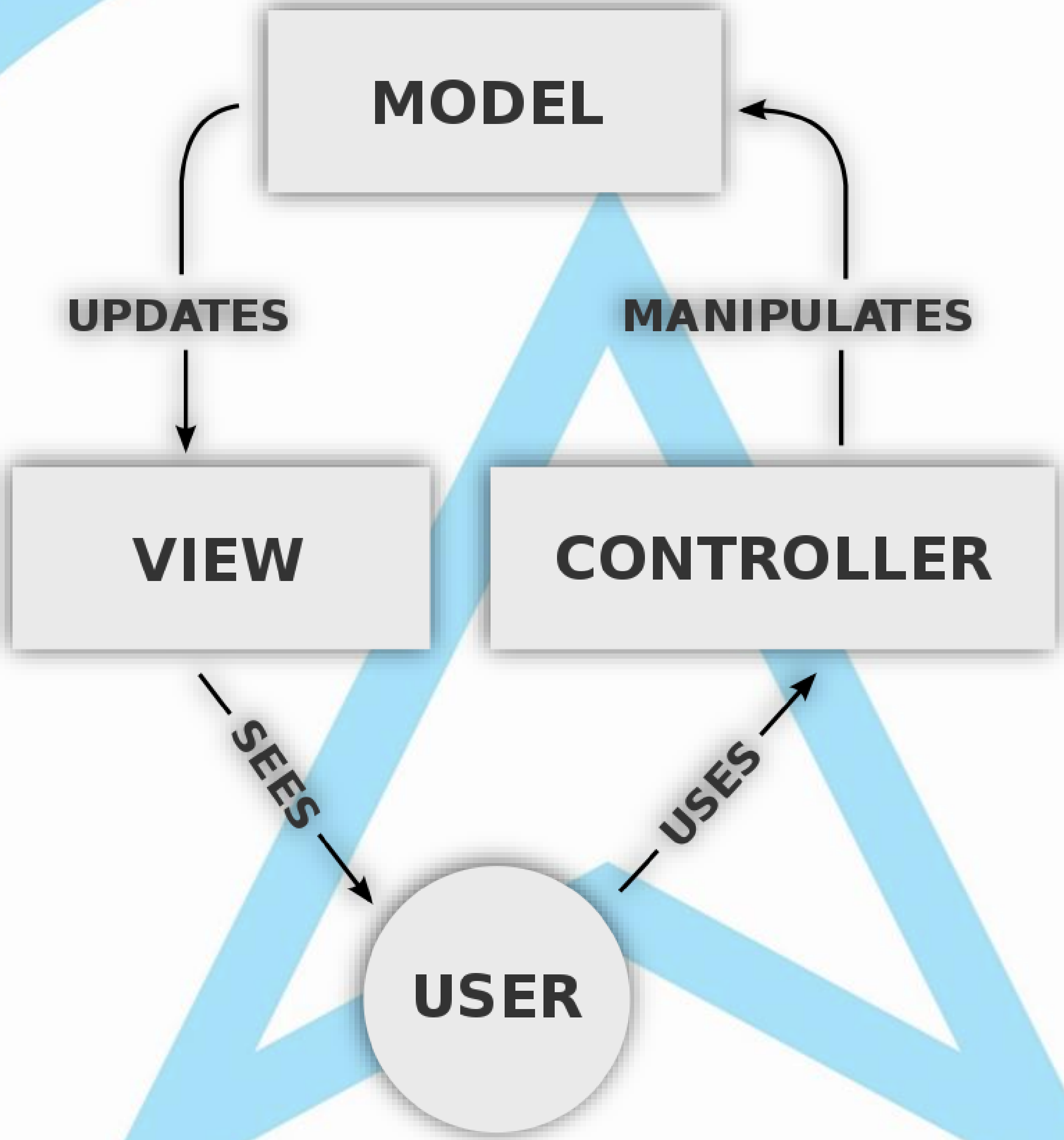
ASP.Net MVC

- Active Server Pages (ASP)
  - Framework for dynamic web applications
  - Initially released in 1996
  - ASP.Net released in 2002
  - ASP.Net Core first released 2016
    - Open-source, cross platform
- Model-View-Controller (MVC)
  - Architectural design pattern used in 1970’s and 1980’s, and formalized in 1988
  - Many frameworks use this design pattern

  - https://en.wikipedia.org/wiki/Design_Patterns
  - https://exceptionnotfound.net/tag/designpatterns/

MVC

- Model
  - Business data
  - POCO
- View
  - Communication with user
  - Presents model in different ways
  - “Knows about” Model
- Controller
  - Glue / mediator / orchestrator
  - Gets model data, selects view
  - “Knows about” Model, View



ASP.Net MVC Project (Visual Studio)

- Create a project
- Project structure
  - Dependencies
  - wwwroot
  - Controllers
  - Models
  - Views

Let’s Code

Default Routing

- http://localhost/{controller}/{action}/{id?}
- Framework looks for a controller called {controller}Controller
  - E.g. HomeController
- Looks for an Action (method) on that class called {action}
  - E.g., Index
- Passes {id} as a parameter to the Action (optional)
- Action (method) returns a View (html)
  - Might also return a file (e.g., CSV), JSON, XML or other data (later)
  - Technically, it returns any *IActionResult*
- Default values
  - {controller} defaults to Home
  - {action} defaults to Index
- Routing is setup in Startup.cs

Views

- HTML that is sent to the user
- Razor
  - HTML / C# hybrid (.cshtml file)
  - @ delimits “server code”
- Blocks: @{ }
- @if
- @for
- Creating a new view
- Layout Views

Let’s Code

Layout View

- Each view has a Layout property
  - May refer to a Layout file
  - Layout file supplies “common” elements across pages
  - View renders “within” the Layout
  - _ViewStart.cshtml supplies the “default” value for Layout
- Typically in Views\Shared folder
- Called _Layout.cshtml by default
  - Contains a @RenderBody() statement
  - That’s where the individual view content goes

Let’s Code

ASP Attributes

- **<a asp-area**="" **asp-controller**="Home" **asp-action**="Index">Home</**a**>
  - Generates

  <a href="/">Home</a>
- **<a asp-area=**"" **asp-controller**="Home" **asp-action**="Privacy">Privacy</**a**>
  - Generates

  <a href="/Home/Privacy">Privacy</a>

Let's
Code

A little more on ASP Attributes

- <**a asp-area**="" **asp-controller**="Home" **asp-action**="Index">Home</**a**>
  - Generates <a href="/">Home</a>
- <**a asp-area**="" **asp-controller**="Home" **asp-action**="Privacy">Privacy</**a**>
  - Generates <a href="/Home/Privacy">Privacy</a>

- <**a asp-controller**="Accounts" **asp-action**="Details" **asp-route-id**="5">Details</**a**>
  - Generates <a href="/Accounts/Details/5">Details</a>
- <**a asp-controller**="Accounts" **asp-action**="List" **asp-route-sort**="Balance">Accounts</**a**>
  - Generates <a href="/Accounts/List?sort=Balance">Accounts</a>

Let’s
Code