**Topic: Into to Web Development**   
(Previous topic: None)

* Programming best practices
  + Minimize dependencies
  + Separation of concerns
  + Unit tests
* System Architectures
  + Three tier
* UI Architectures
  + Smart UI (Razor Pages)
    - Look at a past project
    - .cshtml (form) and .cshtml.cs (code behind)
    - Identify dependencies
    - Does the pattern enforce any “best practices”?
  + MVC
    - Not new, originated in 1978 with the Smalltalk project at Xerox PARC
    - Became popular with Ruby on Rails, 2005
    - Draw a simple block diagram: M  
       ^ ^  
       / \  
       V 🡨 C
    - Define:
      * Model (data),
      * View (generates HTTP response),
      * Controller (responds to HTTP request)
    - Embraces the statelessness of HTTP
      * No false state implementation
      * No complicated page lifecycle (handling click events)
    - View gives you full control of HTML
* Build an MVC Movie web site
  + Identify dependencies
* Syllabus