



# WEB 16

Inlämningsuppgiften

# Kursen

- Webben, intro
- HTML
- CSS
- Bootstrap
- Javascript
- Http
- ASP.NET
- Razor Pages (MVVM)
- DB repetition
- Identity
- Azure
- MVC och API
- WebAPI och REST (Integration mot externa tjänster)
- Blazor

# Betygskrav

- Godkänt
  - Den studerande kan
    - självständigt bygga, utveckla och felsöka en webapplikation i ASP.NET med hjälp av frontend ramverk utifrån en beställares behov
    - utveckla REST-tjänster (webb API)
    - självständigt integrera en app mot ett externt API
    - självständigt publicera en webapp tillsammans med en databas i Azure.
- Väl Godkänt (VG)
  - Den studerande visa förmågan att lösa programmeringsrelaterade uppgifter på ett genomtänkt sätt som påvisar djupare förståelse för kodens uppbyggnad.

# Inlämningsuppgiften

- Du väljer **ett av** projekten, eller föreslår ett eget projekt
  - När du valt projekt, lämna in ditt val på Moodle,
- Du väljer om du gör G-delen enbart, eller även VG-delen.
  - Om du väljer VG-delen, se till att G-delen är klar först.
- Uppgiften är tänkt att genomföras enskilt.

# Inlämningsuppgiften

- Krav
  - **Razor Pages**, eller MVC (motivera varför)
  - Minst en del av applikationen ska funktionellt vara frikopplad från huvudprojektet (helt eget projekt), och exponeras ut som ett API
  - Publicerat på Azure
  - All kod ska följa Dependency Injection-mönstret
  - Bootstrap-fint (men med egen CSS också)
  - Fler krav kan komma (agilt)

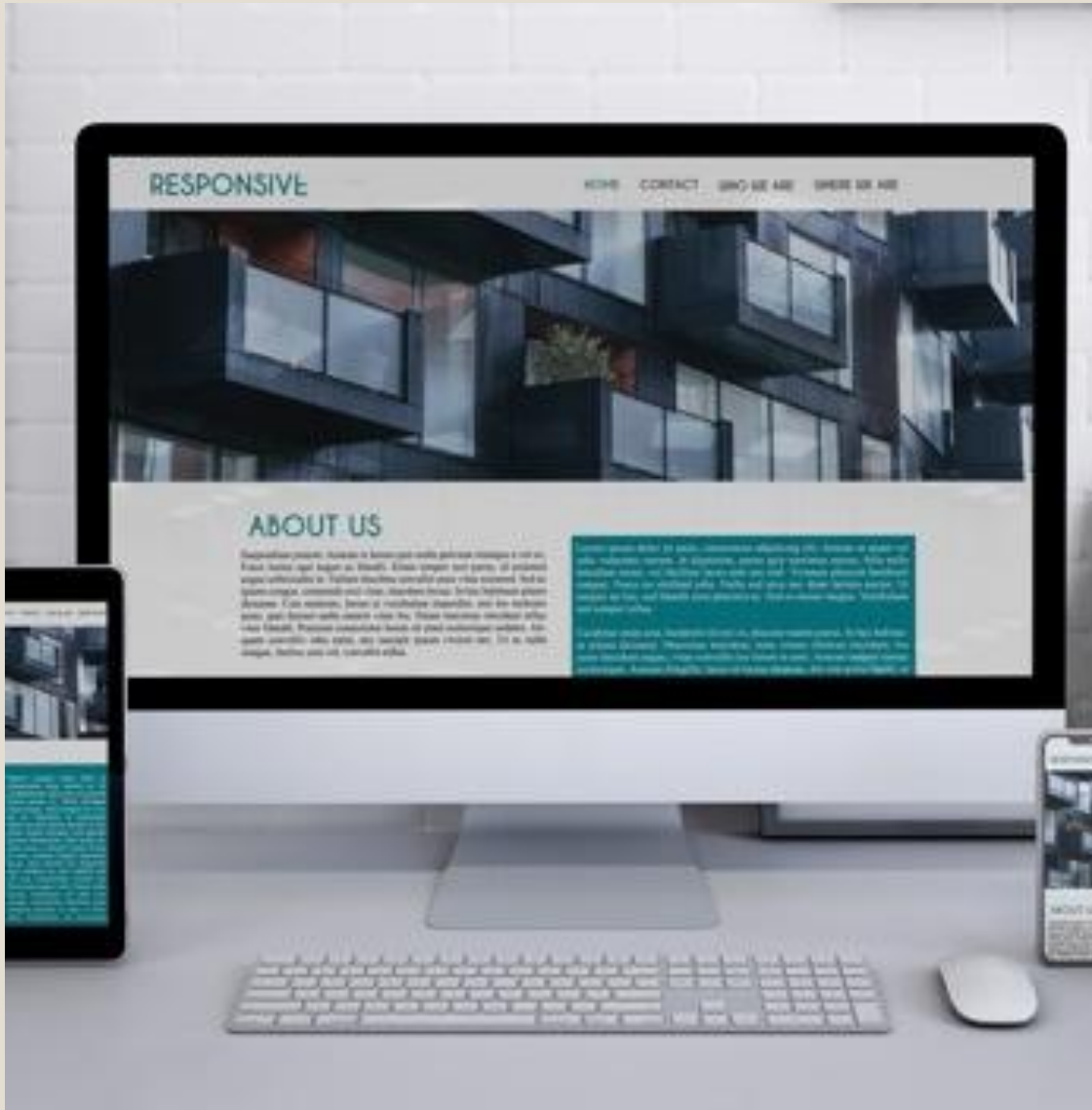
# Arbetsgång

1. Välj vilken uppgift du vill göra.
  - Meddela läraren ovanstående beslut
2. Fråga läraren om något är oklart
3. Fundera på vilka saker som behövs för uppgiften
  - Teknik, tid, kunskaper...
4. Bestäm vilka projektverktyg du behöver
  - Kanban-tavla
  - SCRUM (för grupper)
5. Börja göra skisser på papper, hur sajten ska se ut.
6. Fundera på vilka modeller/klasser som behövs
7. Börja arbeta

# Inlämningsuppgift

## - Public

- Publiceringssystemet Public
  - Ett publiceringssystem eller CMS (Content Management System) används som grund för en hemsida. I systemet styr du som användare enkelt innehållet på din hemsida, lägger till sidor, uppdaterar texter, lägga in bilder, dokument och andra filer - allt detta helt utan kodkunskaper.
- I den här uppgiften ska du bygga ett Publiceringssystem som är enkelt att använda, för någon utan tekniska kunskaper.



# Inlämningsuppgift - Public

- Exempel på publiceringssystem

- Umbraco CMS ([Umbraco - the flexible open source .NET CMS](#))

- Systemet är helt gratis (open source) och körs på Microsoftplattformen ASP.NET. Nyligen valde Microsoft att supporta detta projekt vilket är en tydlig vägvisare om att Umbraco är och förblir en stor aktör på marknaden. Systemet används idag på över 80.000 webbplatser och av flera internationella företag. En fördel med att välja Umbraco är att det finns mängder med professionella utvecklare både i Sverige och övriga Europa.

- EPI Server CMS ([Episerver: The World's Leading Digital Experience Platform | Episerver](#))

- Ett ASP.NET-system utvecklat i Kista utanför Stockholm. Detta publiceringssystem används bland annat av Elektrolux, Arla, Sas, Pfizers och många andra svenska företag och myndigheter. EPI Server är marknadsledande i nordnorden och har 100-tals partnerföretag som utvecklar siter på denna plattform.

- Wordpress ([WordPress.com: Create a Free Website or Blog](#))

- Började som en bloggmotor men har på senare år utvecklats till ett system som många väljer att använda som grund till enklare webbsidor. Fördelarna med Wordpress är många och systemet passar perfekt till mindre hemsidor.

- Drupal ([Drupal - Open Source CMS | Drupal.org](#))

- Även detta ett system som bygger på öppen källkod (Open Source) vilket gör att utvecklare från hela världen kan skriva kod till det. Drupal är byggt i linux-miljö med PHP som språk och MySQL som databas, detta gör att driftkostnaden är lägre än liknande system som är byggt på Microsofts servertekniker. Till drupal finns många plugin-moduler som ofta är helt gratis.



# Inlämningsuppgift – Public – G

- Följande funktioner ska finnas:
  - Inloggning för admin
  - Skapa meny för sajten, samt kunna flytta och radera menyalternativ.
  - Skapa sidor (En sida är inte en ny cshtml-fil, utan en växling av innehåll, på samma sida)
  - Möjlighet att lägga in innehåll på sidan, i main-delen:
    - Rubrik
    - Text (Samma utseende på hela texten)
    - Bilder
    - Länkar
  - Enkelt admin för att bestämma hur en rubrik, text, bild, länk ska se ut, via CSS.
  - Besöksräknarfunktionalitet, antal besökare per sida.
  - Responsiv design, fungera i mobiltelefon

# Inlämningsuppgift – Public – VG

- G-delen plus:
  - Möjlighet att formatera delar av texten, t ex fet stil, olika typsnitt osv
  - Möjlighet att kunna lägga till innehåll på flera ställen på sidan (t ex vänsterkolumn, footern)
  - Möjlighet att infoga plugins, i eget definierat format, som är helt obundet till Public kodmässigt.
    - Exempelvis:
      - Visa besöksräknare
      - Bildgalleri
      - Väder
  - SEO-anpassad URL/URI för sajtens sidor
  - Det finns utrymme för egna ideér och funktioner.



# Inlämningsuppgift

## – Snackis

- Ett diskussionsforum
  - Internetforum, är webbaserade diskussionsforum på internet.
  - Ett forum har vanligen en hierarkisk eller träd-liknande struktur, med huvud- och underkategorier.
  - Beroende på forumets inställningar så kan användare vara anonyma eller behöva registrera sig för att kunna använda det. På de flesta forum behöver man inte logga in för att kunna läsa diskussioner, däremot för att skriva något.

# Exempel på forum

- [Forum \(sweclockers.com\)](http://sweclockers.com)
- [Sporthoj.com](http://Sporthoj.com)
- [Byggahus forum](#)
- Samling med länkar: [Svenska Forum - Lista över dom största svenska diskussionsforumen](#)

# Inlämningsuppgift – Snackis - G

- Följande funktioner ska finnas:
  - En admin ska kunna skapa Rubrik på forumet och övergripande ämnen, såsom
    - Fordonssnack
      - Bilsnack, Båtsnack, Cykelsnack
  - Vem som helst ska kunna registrera sig, och kan därefter skapa ett eget inlägg, under en av de övergripande ämnena.
  - Användaren ska också kunna ladda upp en bild på sig själv, som syns vid sidan om inlägget.
  - Vem som helst ska kunna kommentera inläggen, och allt sker i en enda lång tråd.
  - Varje inlägg ska markeras med Datum och namn på författaren.
  - Det ska också gå att skicka privata meddelanden till enskilda personer
  - Det ska också finnas en anmälanfunktion, för stötande inlägg, som en admin sedan ska kunna granska i en lista.
  - Det ska gå att hämta alla inlägg för en diskussion, med författare och datum, via API.

# Inlämningsuppgift – Snackis - VG

- Förutom G-delen:
  - Inläggen ska ordnas i en trädliknande struktur, där man kan kommentera enskilda inlägg, som blir en sub-diskussion.
    - Alternativt så ska ett citat från det tidigare inlägget synas.
  - Det ska gå att lägga in bilder i inlägget.
  - Det ska gå att skapa gruppmeddelanden, som bara når de som ingår i Gruppen
    - Inbjudningsfunktion för Gruppen, samt möjlighet för den som skapade Gruppen att kunna radera personer
  - Varje registrerad användare ska också kunna ha en enkel info-sida om sig själv.
  - Det ska gå att ge inlägget t ex tummen-upp, hjärta m m, och det ska gå att se hur många som t ex gillat inlägget.
  - Fula ord-filter. Om inlägget innehåller opassande ord, så ska dessa ersättas med \*\*\*\*\*
  - Det finns utrymme för egna ideér och funktioner.

# Inlämningsuppgift – Eget projekt

- Om du vill göra ett eget projekt:
  - Börja med att beskriva det på samma sätt som uppgifterna i den här PDF-en
  - Skicka beskrivningen till läraren
- Om vi kommer överens om upplägget, så är det bara att köra igång.
- Om det är ett påbörjat projekt, gör ändå en beskrivning, inkl det som redan är gjort.
- För VG: Hög verkshöjd, uppfylla G, och VG-kraven enligt kursplanen.

# Redovisning - Inlämninguppgiften

- Redovisningsdag
  - 11:e juni (Onsdag) och 13:e juni (fredag)
  - Alla uppgifter redovisas, både G och VG.
- Inlämning
  - Länk till sajten på Azure
  - Länk till GIT



# Redovisningsdagen – Ha med detta

- Vilken uppgift/egen uppgift?
- Betygsambition?
- Sprint review/slutpresentation
  - Lagom längd, max 10 minuter
  - Visa den publicerade Azure-webbappen.
  - Guida igenom ett normal flöde för en användare
  - Använd riktiga namn och ord/meningar
  - Visa admin
  - Visa kod

# Idag

Först: Alla övningar!

Sedan: Inlämningsuppgiften:

1. Välj vilken uppgift du vill göra.
  - Lämna in ditt val på Moodle
2. Fråga, om något är oklart
3. Hör av dig med synpunkter på uppgifterna