

# Inlämning: Auktioner.NET, 6Yhp

**Inlämningsdatum** 9 jan av 23.59      **Poäng** 100

**Lämnar in** en länk till webbplats eller en filuppladdning

**Filtyper** zip

**Tillgänglig** 27 dec 2021 kl 0:00–9 jan kl 23.59 14 dagar

Den här uppgiften låstes 9 jan kl 23.59.



## Scenario

Ett större antikvariatbolag "**Gammalt Glimmer**" (GG) har allt för länge väntat med att digitalisera sig. De har under många år haft en erfaren inventarieansvarig, Göran, som är duktig på att organisera och kategorisera alla dom prylar (objekt) som firman sedan säljer på auktion, men som nu snart ska gå i pension, och GG vill göra det enklare för alla som arbetar på bolaget att samverka utan att behöva förlita sig på Görans expertis.

Redan efter årsskiftet vill dom ha en första version av ett digitalt arbetsverktyg för detta på plats, där man snabbt och enkelt ska kunna lägga upp ett inventarie över vad som ska erbjudas på auktion, och hålla koll på vart objektet befinner sig, vem som är säljare osv.

Nu behöver dom snabbt hitta en erfaren webbutvecklare som kan skapa detta och ha det online på kort tid. Av en händelse är Gammalt Glimmers VD, Greta en återkommande kund till *Uppfinnaren* som du arbetade med nyligen, och du har via den vägen blivit uppmärksammad för uppdraget! Greta skriver så här...

Hej! Vad kul att du kunde ta det här uppdraget med så kort varsel! Vi arbetar generellt sett med att köpa in olika antikvariat som vi sedan erbjuder på våra auktioner. Vi är många som via olika kontakter köper in objekt, men idag är det bara Göran som har full koll på hela vårt inventarie, och vi är rädda för att detta kommer fungera dåligt när han snart slutar. Med det här systemet så vill vi att alla som köper in objekt ska kunna lägga in dem i systemet, och att våra auktionsansvariga som jobbar på plats kan gå in och markera dem som sålda eller inte - jag tror att vi får ett bra arbetsflöde på det sättet.

Den som registrerar objekten bör kunna redigera dem så att de hamnar i rätt kategori och får rätt utgångspris, då det är dem som vet ungefär hur mycket objektet är värt - men auktionsansvariga bör vara dem som markerar dem som sålda och anger slutpriset. Vi behöver också spara hur mycket objektet kostade för oss att lägga in, och utgångspriset får såklart aldrig vara lägre än detta.

Vi kategoriserar generellt våra objekt efter deras årtioende och deras användsområde, exempelvis "transport" för bilar och "hushåll" för nyttoprodukter - det vanligaste är dock "dekoration" som vi använder för mycket - ibland har vi olika kategorier för olika högtider och teman, så det kan ändras ganska mycket. Vi måste dock veta årtiondet för våra interna revision och kunna ange en ganska exakt beskrivning över objektet också. Viktigt är också att vi använder ett ID för alla objekt som anges med tre bokstäver och sex st siffror - det är från när vi i början för många år sen endast sålde bilar, men det funkar bra även för tavlor och gamla möbler :)

Första versionen skulle kunna vara en sida där man kan registrera sig som inköpare, loggar in och och kan skapa objekt man köpt in till GG, och optimalt vore om auktionsansvariga kan logga in och lägga in slutpris och status.

Hör av dig om du har några frågor eller funderingar, det ska bli kul att få detta på plats!

Mvh Greta Granit, Gammalt Glimmer AB

## Tips

Börja med att analysera uppgiften noggrant, fokusera på kravlistan nedanför och jämför med brevet från uppgiftsbeställaren tills du har skapat dig en egen uppfattning om hur slutresultatet ska se ut. Det är viktigt att notera att detta är ett väldigt dynamiskt system - du kan inte skapa klasser för varje typ av objekt som ska beställas, men du kan identifiera klasserna från kundens språk och skapa klasser eller fält utifrån det.

Gällande **tester** så finns det här något mer avancerad domänlogik - du behöver inte testa något annat än reglerna för försäljningsobjekt, exempelvis kan du skriva tester för att kontrollera att alla kostnads-uträkningar blir rätt, var noga med att identifiera reglerna och attributen från brevet du fått av VD:n!

Du behöver inte testa din användarhantering, eller köra den särskilt komplex - men för högre betyg bör du i ditt program kunna skilja på olika typer av användare och vad deras rättigheter är.

Se även tipsen på tillvägagångssätt för uppgiften på föregående sida.

**Lycka till!**

## Inlämning & betygsättning

I denna programmeringsuppgift ska du skapa och ladda upp en applikation som uppfyller den nedanstående kravlistan. Applikationen ska skapas i versionhanteringsystemet Git och laddas upp på Github. När du är klar med uppgiften lämnar du in den lösning du skapat som en länk till det repo du skapat på Canvas. Påbörja arbetet genom att följa nedanstående länk till uppgiften på Github Classroom:

[Github Classroom: Auktioner.NET](https://classroom.github.com/a/3kcTmLbz) [\(https://classroom.github.com/a/3kcTmLbz\)](https://classroom.github.com/a/3kcTmLbz)

När du är klar med uppgiften lämnar du in din lösning som en länk till det repo som skapats på Canvas. Efter rättning kommer du få feedback med korta kommentarer för varje rad i poängtabellen. För att ett krav ska vara poänggivande måste även de krav som angetts som kriterium vara uppfyllda.

För betyget **Godkänd** krävs minst 50 poäng från poängtabellen. För betyget **Väl Godkänd** krävs 80 poäng, samt att minst 20 av dessa är från de krav som är angivna med (vg) i poängtabellen. Erhålls åtminstone 25 poäng så erbjuds ett försök att komplettera din inlämning, annars ges betyget underkänd.

Läs [Om inlämning och deadlines](#) för mer information om programmeringsuppgifter.

## Kravlista

Följande kravlista beskriver lösningen som du ska skapa och lämna in, den fungerar som poängtabell för uppgiften:

Nummer	Max.Poäng	Kriterium	Beskrivning	Läranderesultat
1	2		Lösningen ska vid inlämning vara körbar med .NET Core	4

Nummer	Max.Poäng	Kriterium	Beskrivning	Läranderesultat
2	2	1	Applikationen använder ASP.NET för att starta en webbserver som visar html-sidor när den besöks från en webbläsare	4
3	2	2	En "start"-sida finns som visar (korta) instruktioner för hur hemsidan ska användas, enligt valfri utformning	4
4	2	2	En sida för att lista tillgängliga försäljningsobjekt finns, kallad "inventarie".	4
5	10	2	En besökare kan registrera sig som användare på hemsidan, samt logga in och ut ur systemet, detta ska ske via användning av ASP.NET Core Identity.	5
6	5	4 & 5	Endast inloggade användare har tillgång till en sida för att lägga in nya försäljningsobjekt.	5
7	8	6	Via denna sida (krav 6) kan man som inloggad användare skapa ett nytt försäljningsobjekt.	4
8	8	6	Via en sida kan en inloggad användare redigera ett existerande försäljningsobjekt.	4
9	2	4	Alla försäljningsobjekt hämtas samt sparas via Entity Framework Core	4
10	2	9	Entity Framework ska användas med en databaskoppling till en tillfällig databas via metoden "UseInMemoryDatabase" i Startup.cs	4
11	5	9	Ett försäljningsobjekt innehåller den information som kunden önskar i uppdragsbeskrivningen.	4
12	4	11	Försäljningsobjekt är väl representerade i kod utifrån kundens önskemål i uppdragsbeskrivningen.	4
13	8		Minst två lämpliga enhetstester finns för de regler som nämns i kundens uppdragsbeskrivning.	6
14 (vg)	4	5	En inloggad användare kan endast se sina egna skapade försäljningsobjekt.	5
15 (vg)	4	5	En inloggad <u>auktionsansvarig</u> kan se alla försäljningsobjekt, samt redigera om dem som antingen leverande eller kvar i lagret samt ange slutpriset efter att objektet auktionerats ut.	5

Nummer	Max.Poäng	Kriterium	Beskrivning	Läranderesultat
16 (vg)	2		Lösningen ska förutom kod innehålla en fil med namnet "reflections" i formatet md, txt eller pdf	4, 5, 6
17 (vg)	10	16	I reflections-filen under rubriken "Struktur" finns en kort och väl vald motivering till kodstrukturen för "projekt". (1-2 paragrafer)	5
18 (vg)	10	6 & 16	I reflections-filen under rubriken "Säkerhet" finns en kort beskrivning över hur du säkerhetsställt att endast inloggade användare kan arbeta med projekt. (1-2 paragrafer)	5
19 (vg)	10	14, 15 & 16	I reflections-filen under rubriken "Användare" finns en beskrivning över vilka olika typer av användare som finns i applikationen, och hur detta är implementerat med kod. (1-2 paragrafer)	5