

Lab 3 - Webbserver för bokhandel

Vi ska bygga en webbserver som kommunicerar med databasen vi byggde i lab 2, och som låter användarna bläddra igenom vårt sortiment med böcker från webbläsaren. Servern ska utvecklas i Node.js med hjälp av express och pug.

Lab 3 använder sig endast av de obligatoriska tabellerna från lab2. Ni kan alltså lösa även VG-delen även om ni enbart gjort er lab2 enligt G-delen. Det är också tillåtet att låna någon av era kurskamraters databas om ni föredrar det.

Obligatoriska moment (G & VG)

När man surfar till förstasidan på webservern (t.ex. <http://localhost:3000/>) så ska det visas en lista över alla böcker i databasen (boktabellen). Listan ska ha fyra kolumner med headers: *ISBN, Titel, Författare och Pris*. Om er databas är byggd med en many-many-relation mellan böcker och författare så skriv endast namnet på en (1) författare i kolumnen, men lägg till 'm.fl.' efter på de böcker som har mer än en författare.

Vid bedömning läggs inte jättestor vikt på designen i frontend, men använd gärna bootstrap och t.ex table-hover och försök göra så det ser någorlunda snyggt ut.

När man klickar på en rad i listan ska man länkas till en sida som visar utförligare information. Ni ska använda url:en <http://localhost:3000/book/9789100185596> (ISBN för den bok man klickat på) för detta.

På boksidan ska ni visa *minst* fem attribut, t.ex Titel, Författare, Sidor, Publiceringsdatum och Pris. Sidan ska även visa en bild på boken. Bokus har bilder på alla sina böcker så enklast är att länka dit. Exempel: <https://image.bokus.com/images/9789100185596>

Dessutom ska denna sidan lista de olika butikerna från databasen och visa hur många böcker det finns i varje butik. (Alternativt 'ej i lager' om en butik inte har boken inne).

Slutligen ska det finnas en länk som tar oss tillbaka till listan över alla böcker.

Extra moment endast för VG

För er som vill satsa på VG tänker jag att vi lägger på två extra moment på det vi byggt ovan. Jag vill att användaren ska kunna sortera boklistan, samt redigera data om böckerna.

Sortera boklistan

Man ska kunna klicka på de fyra kolumnernas headers för att välja hur listan ska sorteras. Detta vill jag att ni löser genom att varje header är en klickbar länk tillbaka till samma sida fast med en querystring med en parameter *sortColumn* där värdet visar vilken kolumn man valt att sortera på.

Ex. om man klickar på 'Författare' så länkas man till 'localhost:3000/?sortColumn=Authors'. Koden som servar listan måste alltså kolla värdet på den parametern och sortera listan rätt. Om man inte skickar med parametern, eller sätter ett ogiltigt värde så ska listan sorteras efter titel.

Markera gärna den valda kolumnen, t.ex med en nedåtpil, så man tydligt ser vilken kolumn listan är sorterad på. Ni kan använda t.ex <https://fontawesome.com/> för att hitta lämplig ikon.

Redigera böcker

På sidan med information om enskilda böcker vill jag att ni lägger till en länk 'Redigera bok'. Länken ska ta en till sidan: <http://localhost:3000/book/9789100185596/edit> där ni bygger ett formulär där man kan uppdatera Titel, Författare, samt *minst* 3 andra attribut (men inte ISBN).

Formuläret ska vara förifyllt med de uppgifter som finns om boken i databasen, och det ska finnas en knapp 'Spara ändringar' som POST:ar formuläret till webbservern. Backend-koden som hanterar POST:en skriver de nya värdena till databasen och redirectar sedan tillbaka till bokens informationssida.

Författare visas lämpligen i form av en dropdown (<select>) i formuläret, så man enkelt kan välja bland alla författarna i databasen. Om er databas kan ha flera författare per bok kan ni sätta t.ex <select size="5" multiple> för att få en "select list" där flera alternativ är valbara.

Formulär <form> i HTML kan även ha dolda fält (type="hidden"). Ni kan använda det för att skicka med ISBN till servern, så den vet vilken bok som ska uppdateras med det nya datat.

Skriv också en javascript-funktion som körs på klientsidan när man klickar på 'Spara'-knappen. Koden ska validera data i formuläret och om några värden är fel (t.ex. titel saknas, pris är <= 0, eller datum är inte korrekt angivet) så skickas inget till servern, och användaren informeras om vad som är fel så man kan korrigera detta.

Redovisning

Uppgiften utförs enskilt eller i grupper om 2-4 personer. För redovisning lägger ni upp ert projekt på github och länkar dit. Det ska finnas en beskrivning (readme.md) som beskriver vad jag behöver göra för att kunna testa/köra projektet. Om jag behöver en .bak fil med databasen så ska den finnas med också.

Betygskriterier

För godkänt krävs:

- Hemsidan ska visa en tabell med ISBN, titel, författare och pris för alla böcker från boktabellen i databasen.
- När man klickar på en bok ska man komma till en sida med mer detaljerad info.
- Infosidan ska visa *minst* 5 attribut om boken, samt en bild på omslaget.
- Infosidan ska även lista butikerna och deras saldon.
- Det ska finnas en länk på infosidan, tillbaka till hemsidan / listan.

För väl godkänt krävs även:

- Man ska kunna sortera tabellen med böcker genom att klicka på tabellens headers.
- Infosidan ska ha en länk för att redigera boken.
- Redigeringssidan ska ha ett formulär med förifyllda uppgifter (från databasen).
- Formuläret ska ha Titel, Författare, samt *minst* 3 andra attribut.
- Det ska finnas en knapp för att post:a till servern som då sparar till databasen.
- När man klickar *spara* ska formuläret valideras på klient-sidan innan uppgifterna skickas till servern.