

Projektarbete

Steg 1 - Börja här

1. Ni ska skapa grupper av 2 personer om inget annat har bestämts mellan dig och läraren.

MEDDELA LÄRAREN OMEDELBART.

2. PÅ **ONSDAG 2023-04-05** ska ni presentera en idé tillsammans med ett enkelt ERD-diagram (använd <https://drawsql.app/>) som showcasar en tanke/idé om databasstrukturen för mig över ett handledningsmöte.

BOKA HANDLENINGSTID HÄR:

<https://outlook.office365.com/owa/calendar/Tidsbokningcoachmten@nackademin.se/bookings/>

Har ni inte hunnit bestämma er till på onsdagen så ska ni ändå boka handledning - vi kan oavsätt hjälpas åt att göra progress.

Steg 2 - Github classroom för grupparbete

Den **första** som accepterar github-inbjudan ska skapa ett **team**, och de andra ska också acceptera inbjudan men istället trycka på **join** för ett existerande team. För att få gå med i ett team måste man ha pratat med de involverade och fått godkännande.

All kod som skrivs klassas som er egen. Därför så kommer läraren vid slutet av kursen överföra "ownership" av repositoryn till er, eftersom projekten kan bli värdefulla.

Inbjudan: <https://classroom.github.com/a/0JLvXhDb>

Viktig information om inlämningen

- ****Deadline**** 2023-04-27 23:59 (Torsdag 27e april) - BÖRJA NU!
- Kompletteringsdatum annonseras i efterhand
- Projektet får ****också**** göras i django rest framework + valfritt frontend framework, tex. Vue, React, NextJS, NuxtJS
- NI SKA KLONA REPON DIREKT OCH PUSHA COMMITS LÖPANDE, DU SKA ABSOLUT **INTE** KOMMA MED EN ENDA COMMIT PÅ SLUTET!!!!!!

Redovisning & Bedömning

2023-04-27 så ska er grupp presentera arbetet för mig genom att första visa en demo av hemsidan (ingen powerpoint) på ett avslappnat sätt, följt av att vi kollar på koden.

- Tid publiceras på studentportalen
- Ni bedöms och betygsätts **individuellt**
- Blir någon **sjuk** får den redovisa ensam vid ett annat tillfälle
- Ni ska gå igenom koden, prata om den kort så att jag förstår vilka utmaningar ni haft
- Jag kommer ställa följdfrågor och kolla på koden och ställa tydliga tekniska frågor
- Har ni delat upp arbetet ska ni berätta det så att jag förstår vem som borde svara på vilken fråga
- Märker jag att du inte kan svara på grundläggande frågor om koden så leder det till komplettering - detta är hur jag kontrollerar att ni inte GPT-genererar er kod.
- Att **överdrivet** kommentera koden som ett sätt att svara på mina frågor kommer inte funka, det leder till komplettering, tex:

```
Car(models.Model): # create a database table for cars
    name = models.CharField(max_length=100) # create a varchar in the database with max 100 characters
```

Projektbeskrivning

Ni ska utföra ett större projekt med ett webbramverk såsom Django där ni skapar en webbapplikation. Applikationen ska skapa tydligt värde för användaren och bör vara så kreativ som möjligt. Målet är att ni ska få göra något som ni båda i gruppen tycker låter kul som ni blir motiverade att jobba på - det är ERAN IDÉ! **Försök hitta ett gemensamt intresse.**

Exempelvis (men absolut inte begränsat till):

- Gillar du fotboll? Gör en sida med X funktionalitet kopplad till fotboll!
- Gillar du spel? Gör något spelrelaterat!
- Gillar du film, musik eller anime? Do it!
- Har du någon idé kopplad till maskininlärning och tex. openAI med chatGPT? Do it!

Godkänd

Riktlinjer är att applikationen ska ha CRUD-funktionalitet (*CREATE, READ, UPDATE, DELETE*) och en framsida (frontend) som går att interagera, det vill säga:

- Applikationen ska använda dynamisk data från en databas för vissa sidor (READ)
- Applikationen ska låta användare skapa ny data i databasen (CREATE)
- Applikationen ska låta användaren uppdatera befintlig data (UPDATE)
- Applikationen ska låta användare ta bort data (DELETE) om det är logiskt
- Applikationen ska ha en frontend som användare kan interagera med (tex. via django template-system) som implementerar en acceptabel design du skulle vågat visa på en intervju eller github - för att uppnå detta rekommenderar jag ett stylingramverk såsom tailwind eller bootstrap.

Minst 2 av dessa ovanstående punkter ska vara implementerade. Se det som en riktlinje och inte en begränsning, det viktiga är att eran idé gör något intressant.

NI SKA ABSOLUT INTE FÖLJA EN TUTORIAL OCH HÄRMA EN ANNAN IDÉ!!!

Väl godkänd

Du ska implementera minst **två av nedanstående punkter**:

- **Asynkron funktionalitet** (Exempelvis att skicka ett mail utan att användaren behöver vänta eller att utföra scraping utan att användaren bör vänta)
- **Schemaläggning** av datainsamling eller annan funktionalitet som logiskt bör ske
- **Caching**
- **Logging** (kräver extra mycket på denna punkt för att det ska räknas)
- Ett **externt paket** som lägger till funktionalitet som är **nödvändig** för att applikationen ska leverera värde (tex. django-filter, celery, django-storages, django-allauth, django-redis) **osv.** Paket som "django-extensions" eller "debug-toolbar" räknas inte, dem förbättrar er developer experience, dock ingen tydlig funktionalitet (men jag rekommenderar att ni använder dem!).