Eksamen B

I fagene

Programmering og Teknik

2023, juni, kl. 9.00 – 13.00 (14.30)

Alle skriftlige materialer, pc'er, bærbare computere og internet ressourcer er tilladt til eksamen.

Mobiltelefoner og kommunikation med andre personer (inkl. chatbots), bortset fra kommunikation med eksaminator og censor, er forbudt. Dog er det tilladt at bruge GitHub Copilot

Du må ikke gemme dine løsninger på eksterne netværksdrev/værter som GitHub, Facebook o. lign. Overtrædelse af dette vil medføre bortvisning fra eksamen og passende sanktioner vil senere blive tildelt både til afsender/uploader og modtager.

Ved eksamens afslutning skal du lægge din løsning op i Wiseflow.

Eksamen varer 4 timer og efterfølges af 1 times evaluering. I disse tidsrum kan du kun forlade eksamenslokalet med henblik på toilet besøg. Rygning er ikke tilladt.

Ud over disse opgaver, kan du kan blive bedt at besvare nogle ekstra spørgsmål omkring dine svar og eventuelle andre valg.

Kontroller at dette sæt indeholder 3 sider inklusive forsiden.

# Introduktion

I denne opgave skal du arbejde med socket programmering (TCP og UDP), JavaScript og UI-test med Selenium.

Du skal lave forskellige dele af en applikation, der håndterer huse f.eks. hos en ejendomsmægler.

Et hus har flg. Properties (id, address, constructionYear)

|  |  |  |
| --- | --- | --- |
| **Id** | **Address** | **ConstructionYear** |
| 1 | Birkelunden 65, Gunderslev | 1965 |
| 2 | Markskellet 43, Hundelev | 1953 |
| 3 | Strandgade 2, Assens | 1992 |
| 4 | Vibevej 23, Ringsted | 2012 |

# Opgave 1 Web applikation

Du skal lave en web-applikation med HTML, JavaScript, Vue.js, Axios og Bootstrap.

Til denne opgave er der allerede lavet et REST API, som du skal bruge

* Swagger documentation  
  <https://resthouseexam2023.azurewebsites.net/swagger/index.html>
* End point  
  <https://resthouseexam2023.azurewebsites.net/api/Houses>

Din web-applikation skal kunne

1. Vise en tabel med alle huse
2. Give brugeren mulighed for at tilføje et hus
3. Give brugeren mulighed for at slette et hus  
   Bemærk at det ikke er muligt at slette elementerne med Id 1 til 4.

Opgave 2 UI-test med Selenium

Du skal programmere en UI-test med Selenium og C#.

Du skal teste din web-applikation.

De behøver kun teste en af de tre funktioner i web-applikationen.

Bemærk at andre brugere kan anvende REST service samtidigt med dig.

# Opgave 3 TCP-server

Du skal programmere en TCP server i Python. Du vælger selv et port-nummer til din server.

Protokol: Din TCP-server skal kunne modtaget et request med følgende formater

* ADD:Birkelunden 67, Gunderslev:1965  
  Bemærk, at kolon er brugt som skilletegn mellem de 3 elementer.
* STOP

Din TCP-server skal sende et response med følgende formater

* OK, hvis request er ADD med præcist to parametre (adress og constructionYear)
* ILLEGAL ADD, hvis request er ADD og der ikke er præcist to parametre  
  Serveren behøver ikke gemme data.
* STOPPING, hvis request er STOP
* ILLEGAL REQUEST, hvis request hverken er ADD eller STOP

Serveren skal kunne håndtere flere requests sekventielt (efter hinanden), men servere behøver ikke være concurrent.  
Hvis request er STOP, så skal serveren stoppe.

Afprøv din TCP server med socketTest.

# Opgave 4 TCP-client

Programmer en simpel TCP client i Python.

Dit client program skal

1. Spørge brugeren om et input (request strengen, husk koloner :).
2. Sende request til serveren
3. Modtage response fra serveren
4. Udskrive response på skærmen

# Opgave 5 TCP vs UDP

Forklar hvornår du vil bruge TCP henholdsvis UDP.

Skriv nogle stik-ord til din besvarelse.

# Opgave 6 UDP-server

Du skal programmere en UDP server i Python.

UDP serveren skal kunne køre samme protokol som ovenstående TCP-server.

Afprøv din UDP-server med socketTest

# Opgave 7 UDP-client

Programmer en simpel UDP client i Python.

Din UDP client skal kunne det samme som ovenstående TCP client.

# Opgave 8 Web-applikation, flere funktioner

Din web-applikation skal udvides så brugeren får mulighed for at sortere og filtrere huse.