Github Portfolio-Hjemmeside

Praktik Opgave
Uge 6-7,2022

Af:

Ramya Kailashnathan

https://ramyakailashnathan.github.io/CV/

TECHNICAL EDUCATION COPENHAGEN (TEC)

February 2022

Ramya Kailashnathan
Datatekniker-Programmering

Indhold

Ramya Kailashnathan
Datatekniker-Programmering

Indledning

Vi har fået til opgave at lave en online portfolio eller CV på Github. GitHub er, hvor over 73 millioner udviklere sammen former fremtiden for software. Det er meget vigtigt for studerende, der studerer programmering, at vi fremviser vores CV og også vores softwarekode til vores potentielle arbejdsgivere. Vi vil lære at bruge Git, Github, oprette en hjemmeside. Vi vil også lære at bruge HTML til at designe websider. Vi vil også bruge Bootstrap 5.

Opgave Logbog

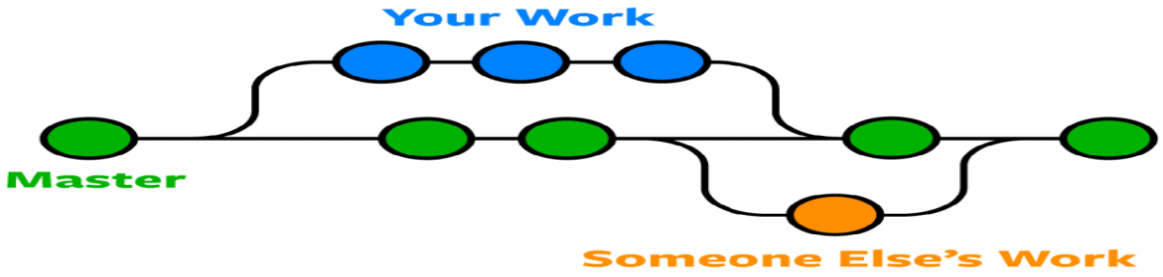
Logbog for opgave										
Uge	06					07				
Dato				10	11	14	15	16	17	18
Dag				Tor	Fri	Man	Tir	Ons	Tor	Fri
Læsning og forståelse af opgaven, video				X						
Svar til 12 spørgsmål						X				
Oprettelse af hjemmeside hjemmeside.io					X					
Oprettelse af html-filen				X						
Formatering af opgave word-dokument						X				

Ramya Kailashnathan
Datatekniker-Programmering

Hvad er Git

1. Du skal forklar hvorfor git findes?

Git er det mest brugte versionskontrolsystem. Git sporer de ændringer, du foretager i filer, så du har en fortegnelse over, hvad der er blevet gjort, og du kan vende tilbage til specifikke versioner, hvis du nogensinde har brug for det. Git gør også samarbejde nemmere, og tillader ændringer af flere personer at blive flettet sammen til én kilde.



Git kan automatisk flette ændringerne, så to personer kan endda arbejde på forskellige dele af den samme fil og senere flette disse ændringer uden at miste hinandens arbejde!

2. Hvad gør git og hvorfor er det brugbart?

Projekter fra det virkelige liv har generelt flere udviklere, der arbejder parallelt. Så et versionskontrolsystem som Git er nødvendigt for at sikre, at der ikke er kodekonflikter mellem udviklerne. Derudover ændres kravene i sådanne projekter ofte. Så et versionskontrolsystem giver udviklere mulighed for at vende tilbage og gå tilbage til en ældre version af koden.

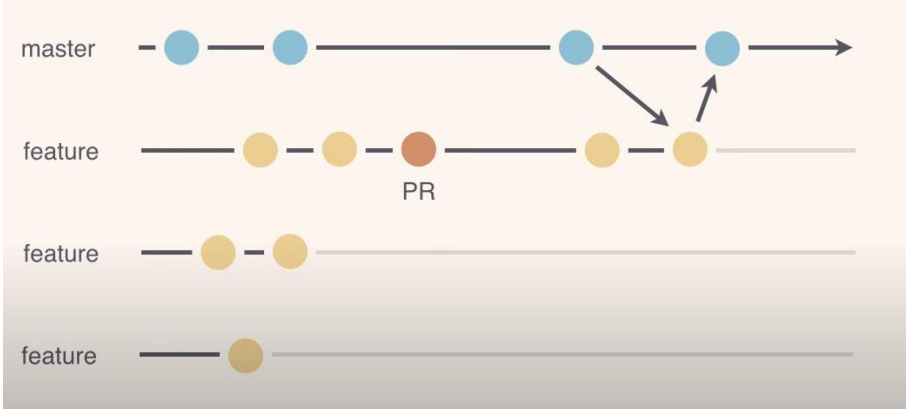
Endelig involverer nogle gange flere projekter, der køres parallelt, den samme kodebase. I et sådant tilfælde er konceptet med forgrening i Git meget vigtigt.

3. Hvordan bruger man git på et projekt, når man er flere udvikler?

Master branch er produktionsklar til enhver tid.

Feature branch ejes af software developer. En enkelt developer kan eje mange funktion branch. Når en funktion udvikles, laver udvikleren en Pull-request (PR i billed), så andre udviklere kan se og teste hans kode.

Nu må du spørge: Hvad er **Pull-request**? Pull-request giver dig mulighed for at fortælle andre om ændringer, du har skubbet til en filial i et lager. Når en pull-request er åbnet, kan du diskutere og gennemgå de potentielle ændringer med andre udviklere og tilføje opfølgningsforpligtelser, før dine ændringer flettes ind i master branch.



a. Sender man projektfilerne til hinanden og arbejder på skift?

Nej, Github tager sig af dette, vi behøver ikke sende projektfiler til hinanden. Github er et centralt lager.

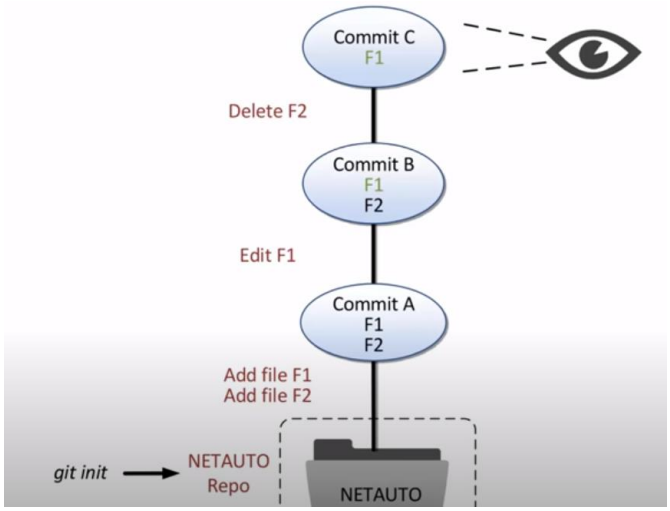
4. Når vi arbejder med git, hvad betyder det så, når vi har et projekt eller Repository?

Git Repository eller projekt er en samling af alle projektfiler sammen med deres historie. Det er en virtuel lagring af dit projekt, hvor du opbevarer alle projektets ressourcer/filer sammen med en speciel mappe kaldet .git .

5. Hvordan arbejder git?

Git giver brugere mulighed for at spore kodeændringer og administrere deres projekt ved hjælp af simple kommandoer. Hjertet i Git er et lager, der bruges til at indeholde et projekt. Et lager kan gemmes lokalt eller på et websted, såsom GitHub. Git giver brugerne mulighed for at gemme flere forskellige repositories og spore hver enkelt uafhængigt.

6. Hvad er et *Commit*?



I Git er en commit et øjebliksbillede af dit lager på et bestemt tidspunkt.

På dette billede kan du se, at vi først har oprettet et repository med **'git init'** kommando.

Efter dette har vi tilføjet F1 og F2 fil. Når vi laver en første commit (**Commit A**), er F1 og F2 gemt i denne commit.

Nu redigerer vi F1-filen og foretager en commit igen, endnu et snapshot tages med en modificeret F1-fil og filen F2.

Dette snapshot er **Commit B**. Derefter sletter vi F2-filen og foretager en commit igen. Dette snapshot er **Commit C**.

Hver commit er således et "**change point"** eller "**save point**". Det er et punkt i projektet, du kan gå tilbage til, hvis du finder en fejl, eller ønsker at lave en ændring.

Bedste praksis – Når vi forpligter os, bør vi altid inkludere et budskab.

7. Hvad er en *Branching* og Hvad er Masten?

b. Hvad er Main?

c. Hvad er forskellen?

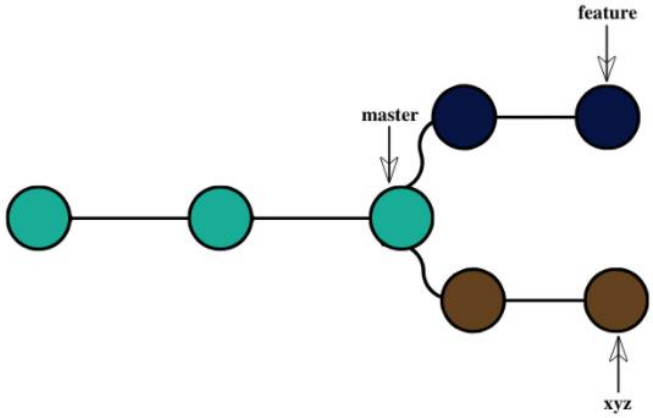
Når arbejdet er afsluttet, kan en gren sammenlægges med hovedprojektet. Du kan endda skifte mellem filialer og arbejde på forskellige projekter, uden at de forstyrrer hinanden.

Hvad er hovedet? Hvad vi forskellen?

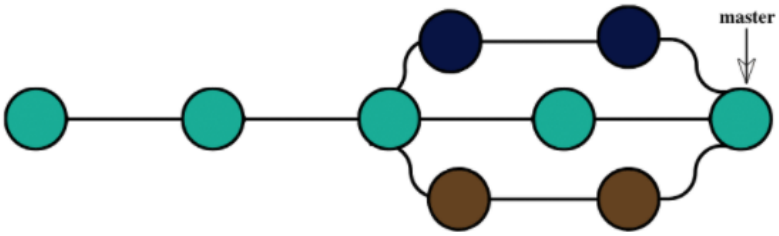
Lad os assume, at du arbejder på et projekt sammen med din ven.

I arbejder begge på to forskellige funktioner og arbejder derfor på to forskellige grene.

Vi kan se det på billedet nedenfor. Feature og xyz er to forskellige functioner.



Når I begge er færdige med arbejdet med jeres særlige funktioner, blev disse funktionsgrene flettet ind i mastergrenen og accepteret i den primære stabile kode.



8. Hvad er *Merging*?

Git merge-kommandoen lader dig tage de uafhængige udviklingslinjer skabt af git branch og integrere dem i en enkelt branch.

9. Hvad er *git Clone?*

Git-kommando - git clone opretter en klon/kopi af et eksisterende lager til en ny mappe.

Det er den mest almindelige kommando, som giver brugerne mulighed for at få en udviklingskopi af et eksisterende centralt lager. Den får adgang til depotet via en ekstern URL.

10. Hvorfor bruger man som udvikler versionskontrol ->> git?

Versionskontrolsoftware som git holder styr på hver ændring af koden i en speciel type database. Hvis der begås en fejl, kan udviklere skrue tiden tilbage og sammenligne tidligere versioner af koden for at hjælpe med at rette fejlen og samtidig minimere forstyrrelser for alle teammedlemmer

11. Forklar om Developer Workflow med git – hvordan det forløber?

Meget enkelt, der er ét centralt lager. Hver udvikler kloner repository, arbejder lokalt på koden, opretter en commit med ændringer og skubber den til det centrale lager, som andre udviklere kan trække og bruge i deres arbejde.

Filialer er selvstændige "spor" af projektudvikling. For hver ny funktionalitet bør der oprettes en ny filial, hvor den nye funktion udvikles og testes. Når funktionen er klar, kan grenen flettes til mastergrenen, så den kan frigives til produktionsserveren.

Her er der to udviklere DEV-30 og DEV-45, der arbejder på nogle funktioner i softwaren.

DEV-45 laver en klon af masteren (vaniljeversion), og begynder at arbejde på funktionen og inkorporerer ændringerne i masteren. DEV -30 kloner og fortsæt med at arbejde selvstændigt.

12. Vis visuelt hvordan Master og flere branchs vil se ud for et givet projekt.

Lad os bygge et hus:

1.Vi har 4 vægge - Vores **Master branch**.

2.Ejeren ønsker en stue og et køkken - **to Feature branch**

3.Jeg vil gemme mit design lokalt - **git commit**. Committing er som at gemme dine ændringer lokalt.

4.Jeg vil gemme mit design et sikkert sted, på repository **- git push**.

5. Development branch & merge request - Vores udviklingsgren er et sted for integration af vores lokaler (eller funktioner).

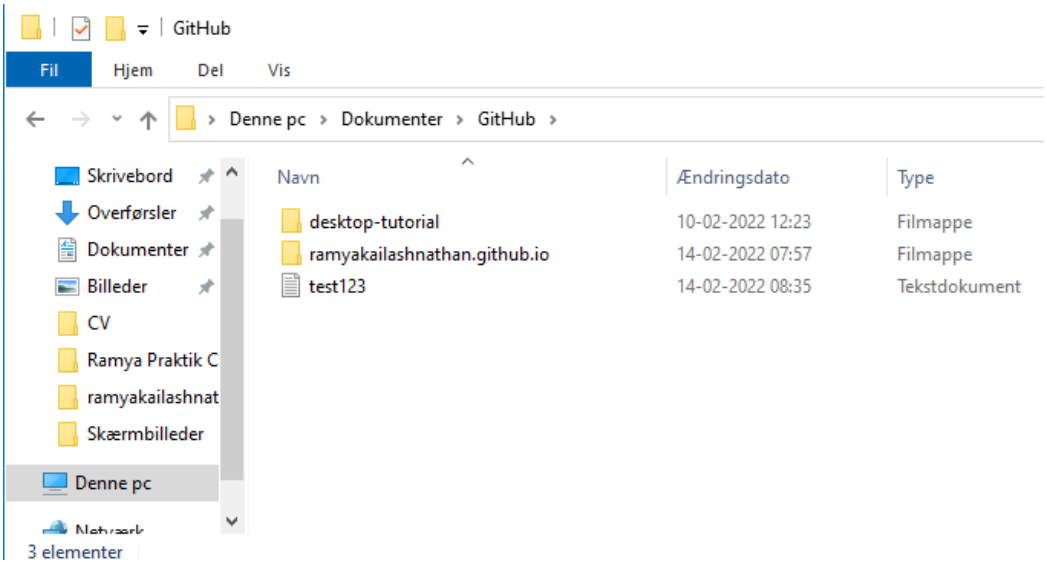
6. Udviklernes design stemmer ikke overens. – **merge conflict**.

7. Afslutningsvis er versionskontrol kernen i nem og sikker softwareudvikling.

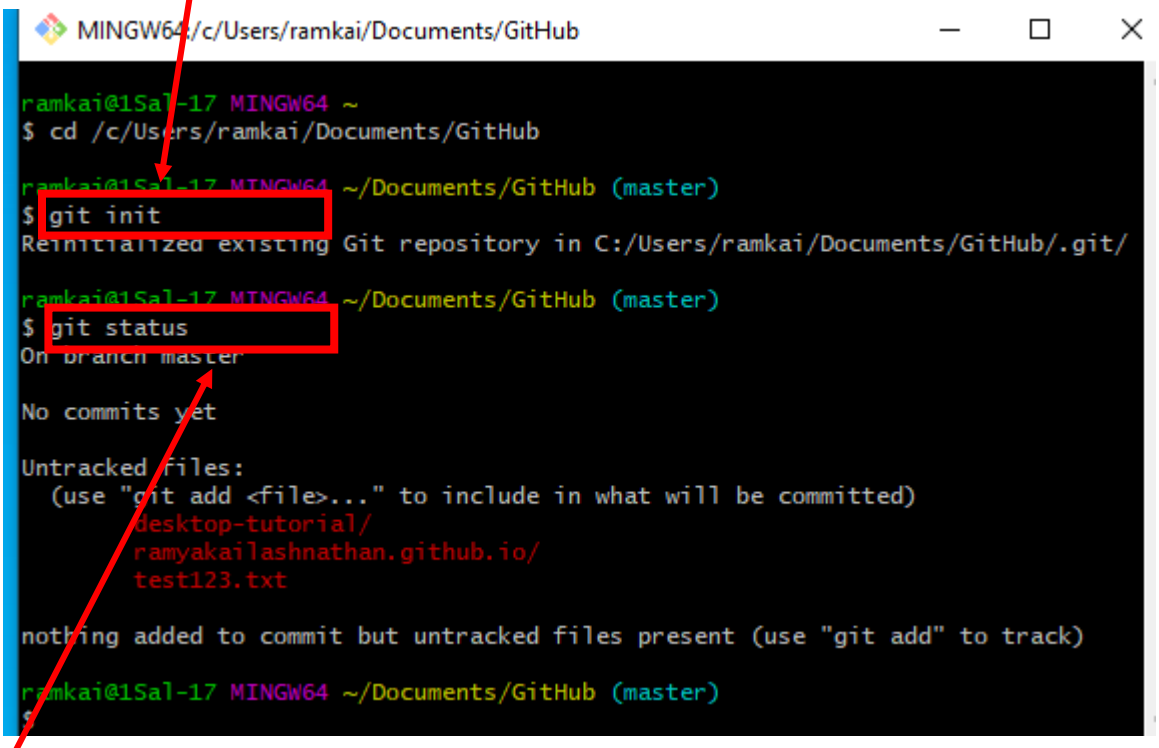
Projekt med Git Bash

1. Dokumenter hele processen. Fra oprettelse til hvordan man laver et *commit* for en Readme.txt fil.

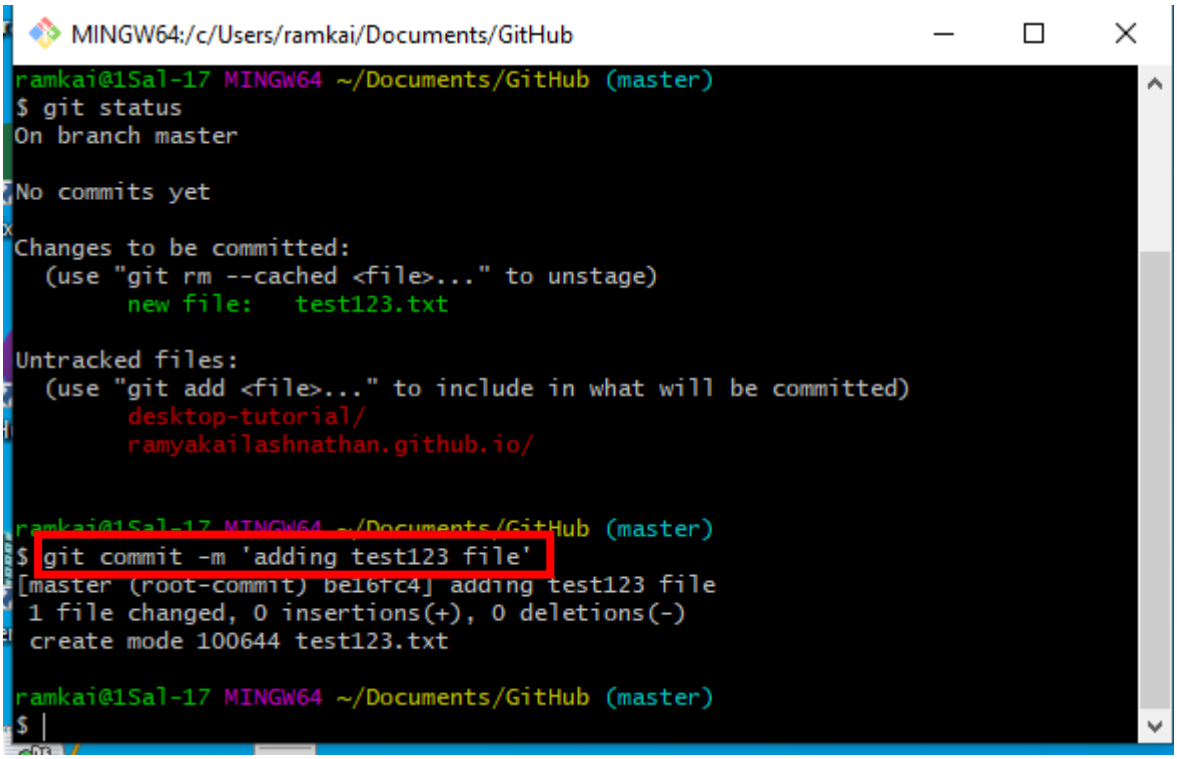
   Jeg har brugt test123.txt .



Du bør først bruge **git init** til at generere et nyt og tomt git-lager.



Git status-kommandoen viser tilstanden for working directory og staging area.

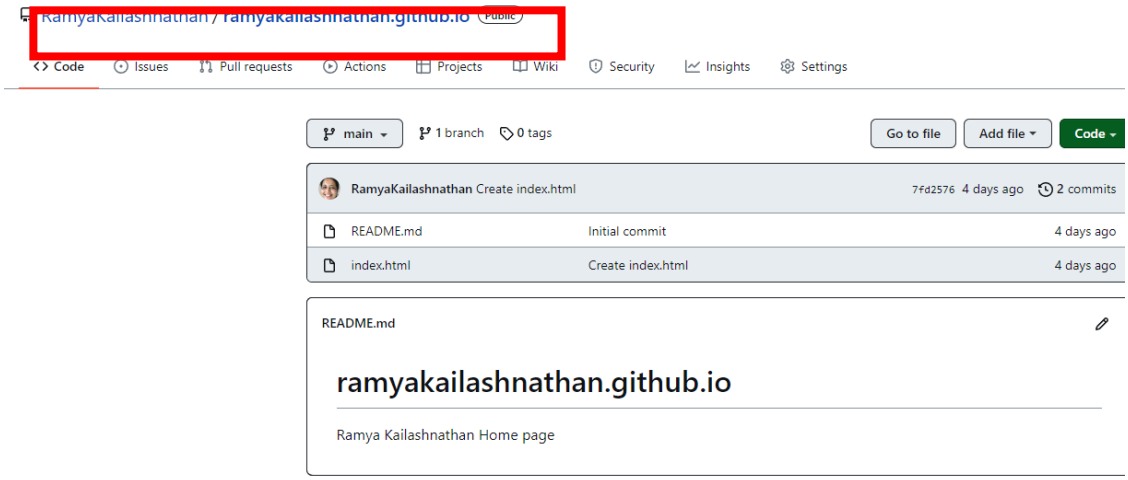


2. *Hvad er en merge conflict?* Hvordan løser man en *merge conflict?*

En merge-konflikt er en hændelse, der finder sted, når Git ikke er i stand til automatisk at løse kodeforskelle mellem to commits. Git kan kun flette ændringerne automatisk, hvis commits er på forskellige linjer eller brancher.

Merge –abort bestået med git log-kommandoen producerer en liste over commits, der er i konflikt mellem flettegrene. Det hjælper med at identificere konfliktfyldte filer. Det hjælper med at finde forskelle mellem tilstandene i et depot eller en fil. Det bruges til at forudsige og forhindre merge konflikter.

Konto på Github.com via GitHub.io

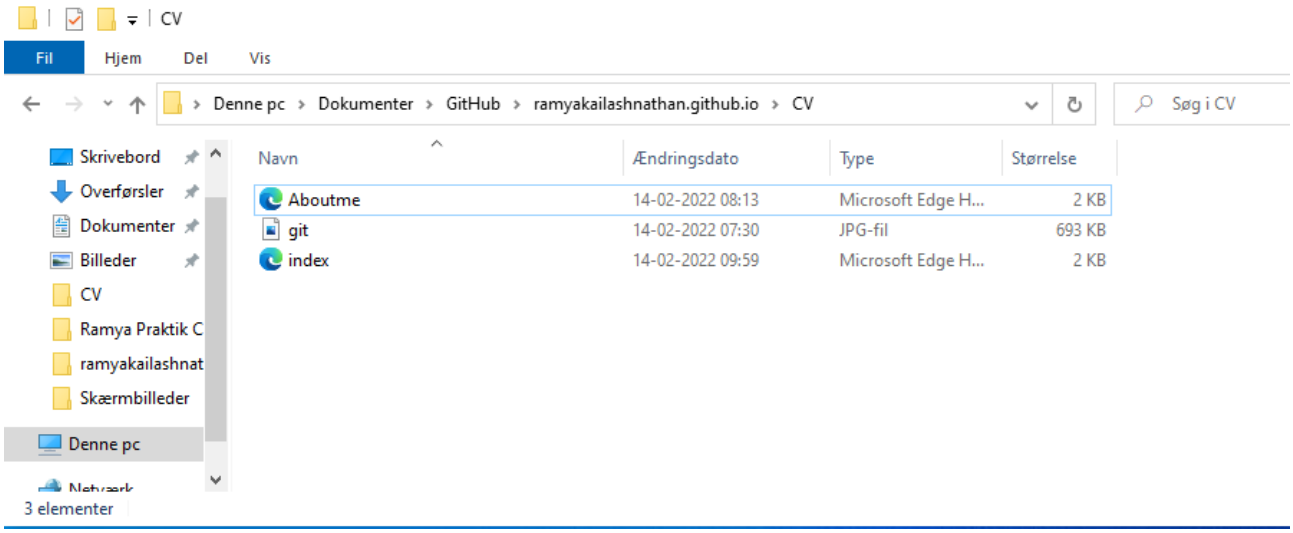


Projekt i Visual studio Code (VS Code)

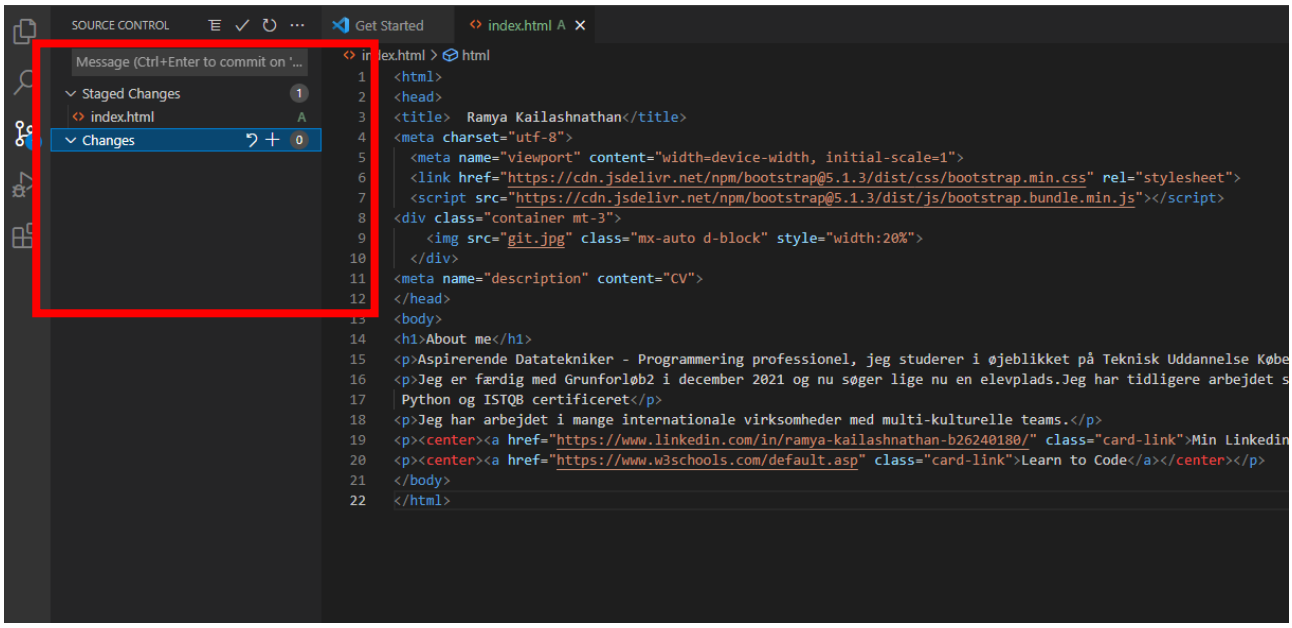
```
<html>
<head>
<title>  Ramya Kailashnathan</title>
<meta charset="utf-8">
 <meta name="viewport" content="width=device-width, initial-scale=1">
 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet">
 <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
<div class="container mt-3">
   <img src="git.jpg" class="mx-auto d-block" style="width:20%">
 </div>
<meta name="description" content="CV">
</head>
<body>
<h1>About me</h1>
<p>Aspirerende Datatekniker - Programmering professionel, jeg studerer i øjeblikket på Teknisk
Uddannelse København (TEC) .</p>
<p>Jeg er færdig med Grunforløb2 i december 2021 og nu søger lige nu en elevplads.Jeg har
tidligere arbejdet som software kvalitetssikringsprofessionel med test af bank- og e-
læringssoftware.
 Python og ISTQB certificeret</p>
```

```
<p>Jeg har arbejdet i mange internationale virksomheder med multi-kulturelle teams.</p>
<p><center><a href="https://www.linkedin.com/in/ramya-kailashnathan-b26240180/"
class="card-link">Min Linkedin Profile</a></center></p>
<p><center><a href="https://www.w3schools.com/default.asp" class="card-link">Learn to
Code</a></center></p>
</body>
</html
```

Et simple domain med github.io

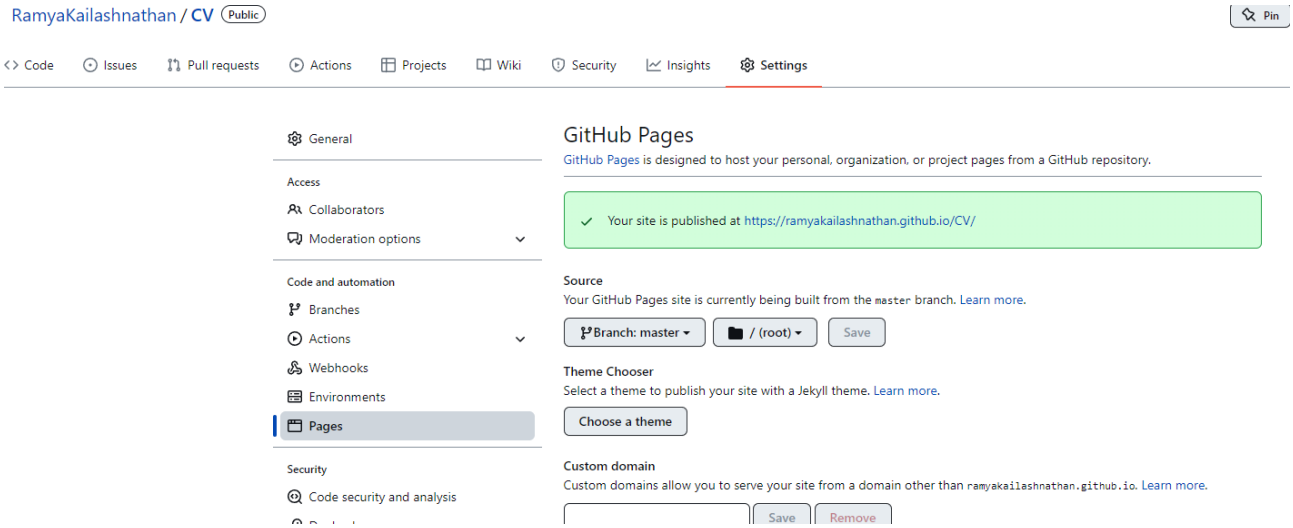


Du kan commit og push thru extensions på Visual code


SOURCE CONTROL
Message (Ctrl+Enter to commit on '...
Staged Changes 1
index.html A
Changes 0
Get Started
index.html A
index.html > html
<html>
<head>
<title> Ramya Kailashnathan</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>

<img src="git.jpg" class="mx-auto d-block" style="width:20%">

<meta name="description" content="CV">
</head>
<body>
<h1>About me</h1>
Aspirerende Datatekniker - Programmering professionel, jeg studerer i øjeblikket på Teknisk Uddannelse Købe
Jeg er færdig med Grunforløb2 i december 2021 og nu søger lige nu en elevplads.Jeg har tidligere arbejdet s
Python og ISTQB certificeret
Jeg har arbejdet i mange internationale virksomheder med multi-kulturelle teams.
<a href="https://www.linkedin.com/in/ramya-kailashnathan-b26240180/" class="card-link">Min Linkedin
Learn to Code
</body>
</html>



Min Github CV

https://ramyakailashnathan.github.io/CV/

Konclusion

Jeg har med succes oprettet hjemmeside. Det er en fantastisk platform til at uploade al min software kode fra nu af.