



Navn på gruppemedlem: Taavi-Topias Henell

## FORSIDE

ved besvarelse av individuell oppgave

<b>Emnekode:</b>	<b><i>UIA IS-114</i></b>
<b>Emnenavn:</b>	<i>Introduksjon til samskaping i informasjonssystemer</i>
<b>Emneansvarlig (faglærer):</b>	<i>Janis Gaillis og Niels F. Garmann-Johnsen</i>
<b>Eventuell veileder:</b>	<i>Tobias Molland</i>
<b>Innleveringsfrist/ tidspunkt:</b>	<i>Fredag 13.sep 2024 23.59</i>
<b>Antall ark inkl. denne forside:</b>	6
<b>Merknader/Evt. tittel på oppgaven:</b>	<i>Git - versjonskontrollsystem.</i>

Eg bekrefter at eg ikkje siterar eller på annan måte brukar andres arbeid utan at dette er oppgitt, og at alle referanser er oppgitt i litteraturlisten.	Ja	Nei
--	----	-----

*Kopiering av andres tekster eller annan bruk av andres arbeider utan kildehenvisning kan bli betraktet som fusk.*

Kan besvarelsen gjenbrukes til forskning og undervisningsformål, og i evt. publiseringer?	Ja	Nei
---	----	-----

# Git - versjonskontrollsystem

Lenke til Repository: <https://github.com/HenellTT/IS-114-Projects>

1. *Kva kan vera argumentar imot å bruka IDE (Integrated Development Environment) som **Visual Studio Code** eller **IntelliJ** om ein er nybegynner i systemutvikling?*

Ved å nytte eit minimalistisk tekstredigeringsverktøy, som Sublime Text, vil nybegynnere ha ein lettere brukeropplevelse. Om ein nyttar til dømes VS Code, kan ein fort bli overvelda av alle moglegheitane og funksjonalitetar til programmet, til dømes ulike extensions, source control, debug console og terminal integrert i VS Code.

2. *Kva er Git og kvifor trenger ein å bruke det? Kvifor er det ikkje anbefalt å installere ein GUI (Graphical User Interface) klient applikasjon for Git?*

Git er eit versjonskontrollsystem som gir utviklere moglegheit til ein sikker og effektiv moglegheit å spore endringar i kildekoden over tid.

Det er ikkje anbefalt å installere ein GUI klient applikasjon for Git, sidan det vil hindre ein forståelse av konsepter innanfor Git som er nyttig å vite. Ved å nytte kommandolinja, vil ein ha betre kontroll og fleksibilitet innanfor kraften til verktøyet.

3. *Kva grunnleggjande kommandoar kan verte nytta på kommandolinja for å kunne sjå filane, som er lagra på eige datamaskin?*

\$ pwd: Viser til kva fil/mappe ein er plassert på/har open på GitBash.

\$ ls -la: Viser til kva fil/mappe er tilgjengelege for lokalisering i den mappa.

\$ cd [lokasjon]: (Change Directory) Endrer lokasjon i mappen.

\$ cd .. : Går oppover / tilbake i mappen

(Cocca, 2022)

*Korleis finn eg dei same filane i **Terminal/GitBash** som eg ser i applikasjonen med grafisk grensesnitt for framvisning av mappe- og fil-struktur på min datamaskin? Kvifor er det anbefalt å læra å bruka Git-kommandoar, som kan verte nytta på kommandolinjen?*

Lokalisere øverste mappen (ofte This PC/c/Users/[bruker]). Ved å nytta kommandoen [\$ cd ..] heilt til ein kommer til /, so er ein kommet til This PC disken. Ofte når ein fyrst opnar terminal/Gitbash er ein ofte allereie på [bruker]-mappa, dermed kan ein nytta kommandoen [\$ ls -la] for å visa dei mappene/filane i denne mappa. Dermed kan ein nytta Change Directory-cmd, til ein kommer til riktig mappe/fil. (Gailis, 2024)

4. *Kva karakteriserer eit godt brukarnamn på Github?*

Å nytta eit brukarnamn som identifiserar ein sjølv eller kva ein jobbar med, som til dømes HenellTT, kan gjera det lettare å identifisera brukaren og deira arbeid.

Kan og vera lettare å hugsa brukarnamnet om det er fengande og lett.

5. *Kvifor treng ein ein slik “token”? Kan ein nytte andre måtar å autentisere seg på Github enn “token”?*

Ein “token” kan defineres som ein nøkkel / id for å identifisera seg sjølv/arbeidet sitt, nesten eit tilleggspassord. For å jobba med GitHub arbeidet sitt er det viktig å identifisere seg sjølv for å unngå at andre har tilgang til det.

Ein kan nytte eksterne third-party applikasjonar som Authenticator-apps, som ein 2 factor authenticator - 2FA.

6. *Kva er ein repository? Kvifor skal eit repositorynamn vera meiningsberande? Kvifor velja ein lisens? Kva er formålet med README-fil og kvifor er det anbefalt å ha det i kvar repository? Kva betyr at repository-en er på Github? Kva betyr det at repository-en har ein grein (branch) som heiter **main**?*

Repository er rett og slett ein arbeidsstasjon for eit prosjekt ein jobbar på, der ein kan lagre mapper og filer, som er til nytte for oppgaven.

Namnet skal vera meiningsberande slik at det er lett for alle å sjå kva prosjektet går ut på, til dømes IS-114-Projects, som er prosjekter innan emnet IS-114.

Når repository er open for offentlig bruk/open kode, må ein velja lisens slik at alle har moglegheit til å nytta, endra og distribuere programvaren.

README-fil nyttar ein til å skriva nyttig informasjon om programvaren, for at andre har moglegheit til å skjønna kva opprettaren har tenkt/framgangsmåten deira.

Når repository-en er på GitHub, betyr det at stasjonen/prosjektet er tilgjengeleg med alle mapper og filer på nettsida GitHub.

Github-branch kan ein tenkja som eit rom der brukaren har arbeidsstasjonen sin på.

7. *Kvifor skal ein kopiere repository-en frå Github til eiga datamaskin? Korleis velgar ein eit eigna sted å kopiera repository-en til på eigen datamaskin? Kva filer og eventuelt mapper som vert kopiert frå Github til eigen datamaskin med kommandoen for kloning?*

Når ein kopierar repository-en frå Github til eiga datamaskin so kan ein gjera og lagre endringar lokalt på eigen datamaskin, utan at den blir endra på Github.

Ein lagrar / kopierar repository-en på datamaskinen i ein plass som ein hugsar lett/finn fram lett i - eg har lagra det i UiA sin OneDrive under eiga IT og Informasjonssystemmappe.

Alle filer og mapper i gjeldande repository vert kopiert frå Github til eiga datamaskin med kommandoen for kloning.

8. *Er den einaste måten å gjera endringane på i filane i ei mappe administrert av Git å åpne eigen applikasjon for tekstredigering? Kunne ein endra andre type filer på same måte, som f.eks. bilde- eller videofiler?*

Den vanlegaste måten å gjera endringar på er å opne eigen applikasjon for tekstredigering, td. VS Code, Sublime eller Notepad. Men ein har også moglegheit å gjera endringar direkte i Github sin webgrensesnitt.

Ein kan endra alle typar filer på same måte. Ved bilde- eller videofiler gjer man:  
\$ gitt add bilde.jpg | \$ git add video.mp4

9. *Korleis finn ein kva er gjeldane mappe på kommandolinja? Korleis treng ein fleire operasjonar for å registrera endringar i Git versjonskontrollsystem? Kan ein klone og kopiera endringane tilbake for ein repository ein ikkje sjølv eig? Kvifor er det nyttig å kopiera endringar gjort i ein fil på eiga datamaskin til Github?*

\$ pwd

For å registrere endringar i Git versjonskontrollsystem gjer man følgjande:

- Lagrar endringane lokalt, dvs. om ein har endra på ein mappe eller ei fil - CTRL+S
- \$ git add \*
- \$ git commit -a "Endringar"
- \$ git push (kan også ha \$ git push origin main)
- Dermed får brukaren ein pull request i GH repository-en sin, og aksepterer denne endringen. (Cocca, 2022)

Ja, ein kan klone og kopiera endringane tilbake for ein repository ein ikkje sjølv eig om ein har link til repository-en, men sjølve pull requests må verta akseptert av eigaren av brukaren.

Ein kan samarbeida med fleire brukarar ved å nytta GitHub. GH sporer alle endringar som vert gjort på repository-en (historikk) og det er enkelt å dele med brukere over heile verda med GitHub.

10. *Kva argumentar kan vera imot å gjera endringar i filane direkte i web-grensesnitt til Github?*

Ein har ikkje moglegheit til å teste ut endringane for filane direkte i web-grensesnitt til Github, noko som kan ha konsekvenser om ein l

11. *Kva respons genererer "push"-kommandoen? Kva er grunnen til ein slik respons?*

[\$ git push] gjev denne responsen  
*Enumerating objects: 5, done.*

Counting objects: 100% (5/5), done.  
Delta compression using up to 12 threads  
Compressing objects: 100% (3/3), done.  
Writing objects: 100% (3/3), 340 bytes | 340.00 KiB/s, done.  
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)  
remote: Resolving deltas: 100% (1/1), completed with 1 local object.  
To <https://github.com/HenellTT/IS-114-Projects.git>  
0e66336..4904731 dev -> dev

Grunnen til ein slik respons er å sjå detaljert kva endringar og kor kjapt dei endringane vart gjort og pusha til repository-en i GitHub

12. Kva kommando vert nytta for sammenslåing? Kvifor må ein redigera fila manuelt?

\$ git merge [branch] [branch]

Det kan oppstå konflikter mellom begge branchene, om begge har same endringar og Git veit ikkje kva endring er riktig

13. Kva er formålet med Github pages?

Github pages gjer det mogleg å hoste nettside som er tilgjengeleg for alle via internett

14. Kvifor er det hensiktsmessig å gjera endringane i den lokale repository-en og ikkje i Github direkte?

Ein har moglegheitar å testa endringar lokalt før ein pusher det til Github direkte. Om ein gjer det på Github direkte, kan ein risikere å gjera noko feil, som kan ha konsekvenser om ein har ein website for repository-en.

15. Kva type kode er denne? Korleis og kvar utføres denne koden?

Det er ein HTML-kode, som står for HyperText Markup Language. Denne koden vert lest av nettlesere og viser resultat av ei nettside.

16. Kva betyr det at ei website er tilgjengeleg på internett?

Ein har tilgang til nettsida/website gjennom smarttelefoner, datamaskiner, ipader osv. til visning om ein er koblet til internett

17. Er denne framgangsmåten, å redigera filer direkte i grenen main, optimal?

Det å redigera ei fil direkte i branchen main er ikkje optimalt. For å gjera det mest optimalt og sikkert, so redigerar ein filane i ein anna branch, td. dev-branch, for å teste det ut på ein sikker plass før ein pusher/merger med main-branches, om ein har td. repository-en oppe som ein website.

18. Kva er fast-forward?

Fast-forward eller "merge" er samanslåing av to brancher/greinar i ein repository.

### 19. "Commit-tre"

Er historikk for alle commits ein har gjort på til Git-repositoryen.

#### *Kjelder:*

Cocca, G. (2022, April 5). Command Line for Beginners – How to Use the Terminal Like a Pro [Full Handbook]. FreeCodeCamp.org;freeCodeCamp.org.

<https://www.freecodecamp.org/news/command-line-for-beginners/> (Cocca, 2022)

Gailis, J. (2024) IS-114-GIT.

[https://github.com/uia-worker/is114komp/blob/main/docs/draft\\_git.html](https://github.com/uia-worker/is114komp/blob/main/docs/draft_git.html) (Gailis, 2024)