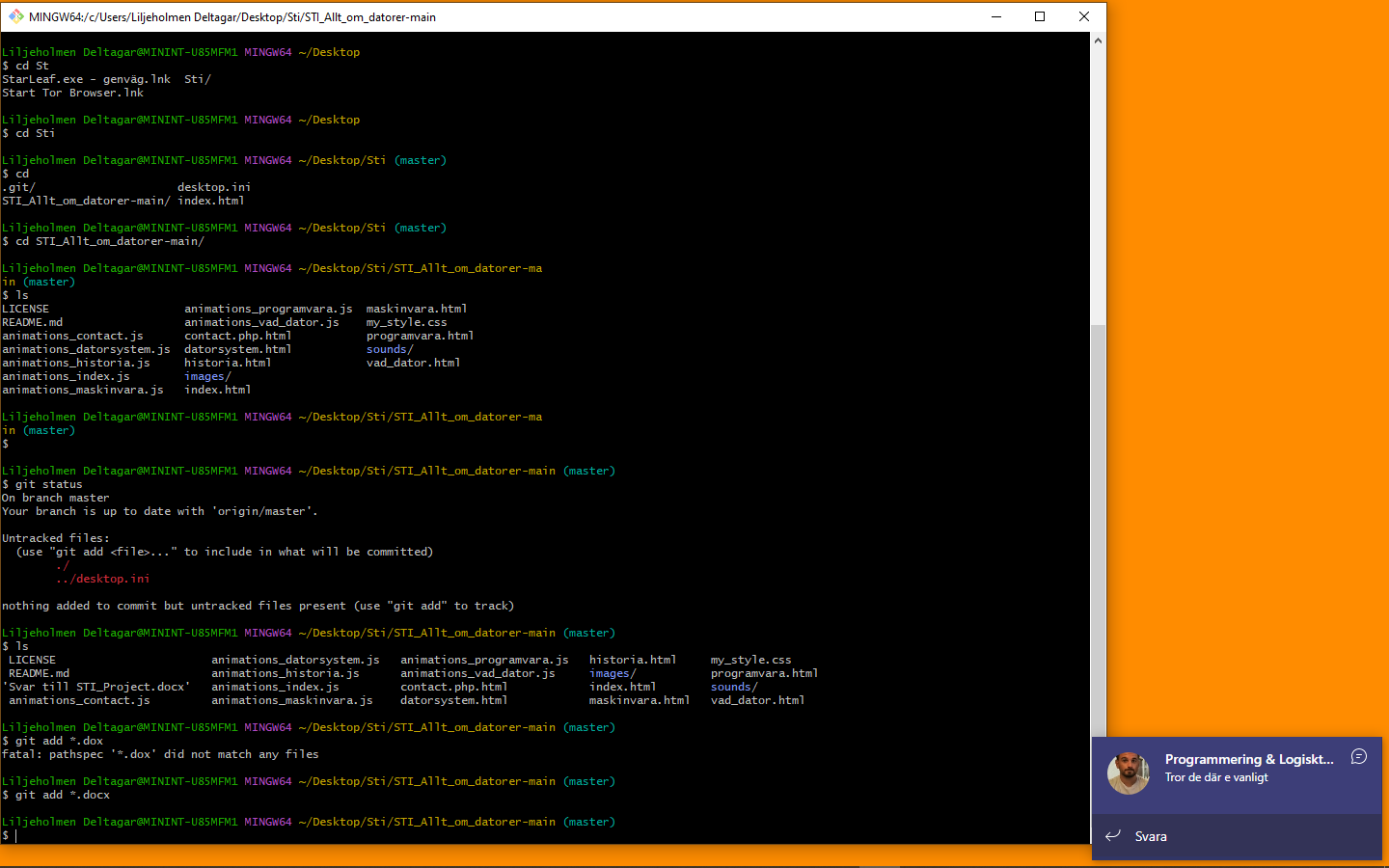
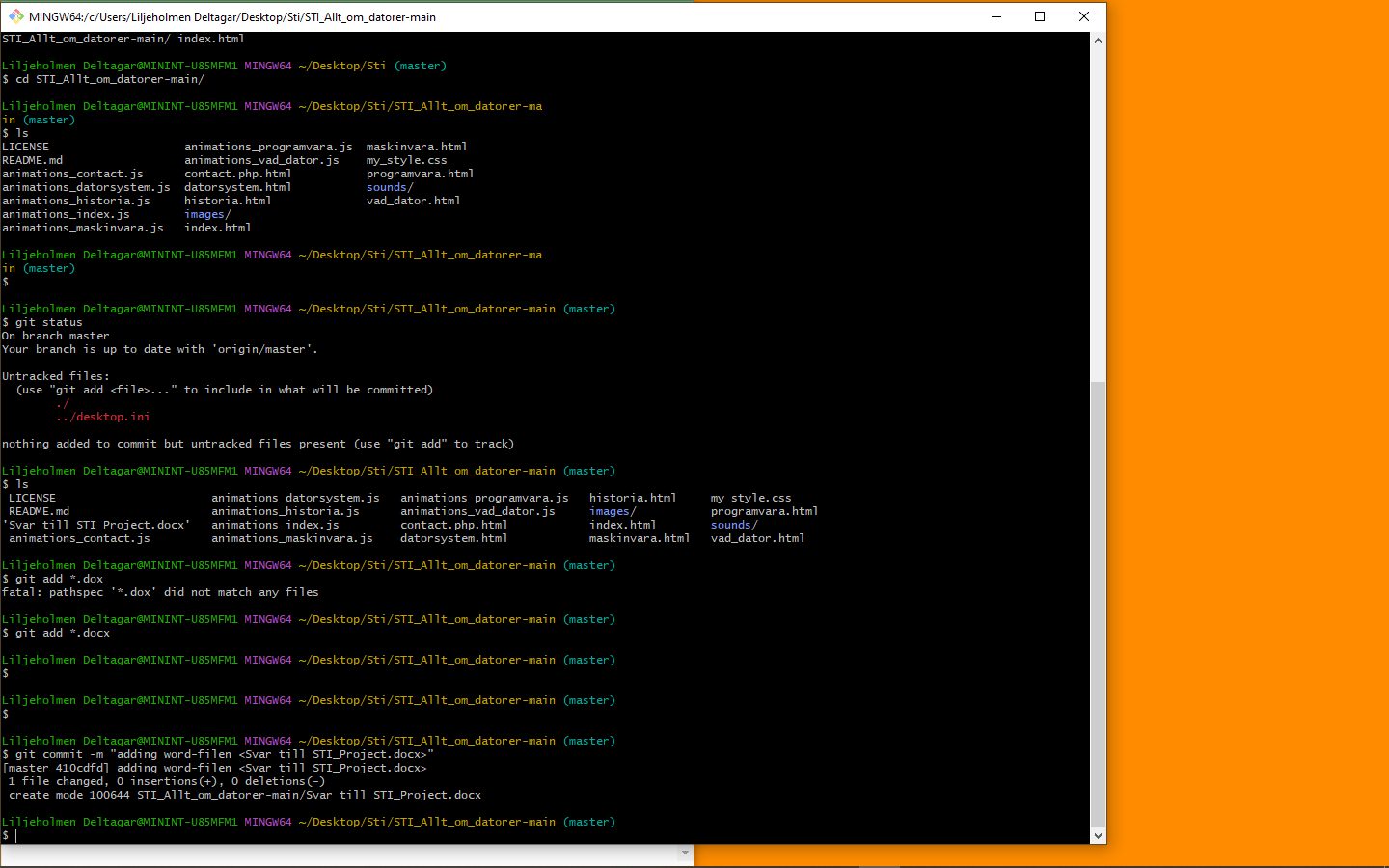
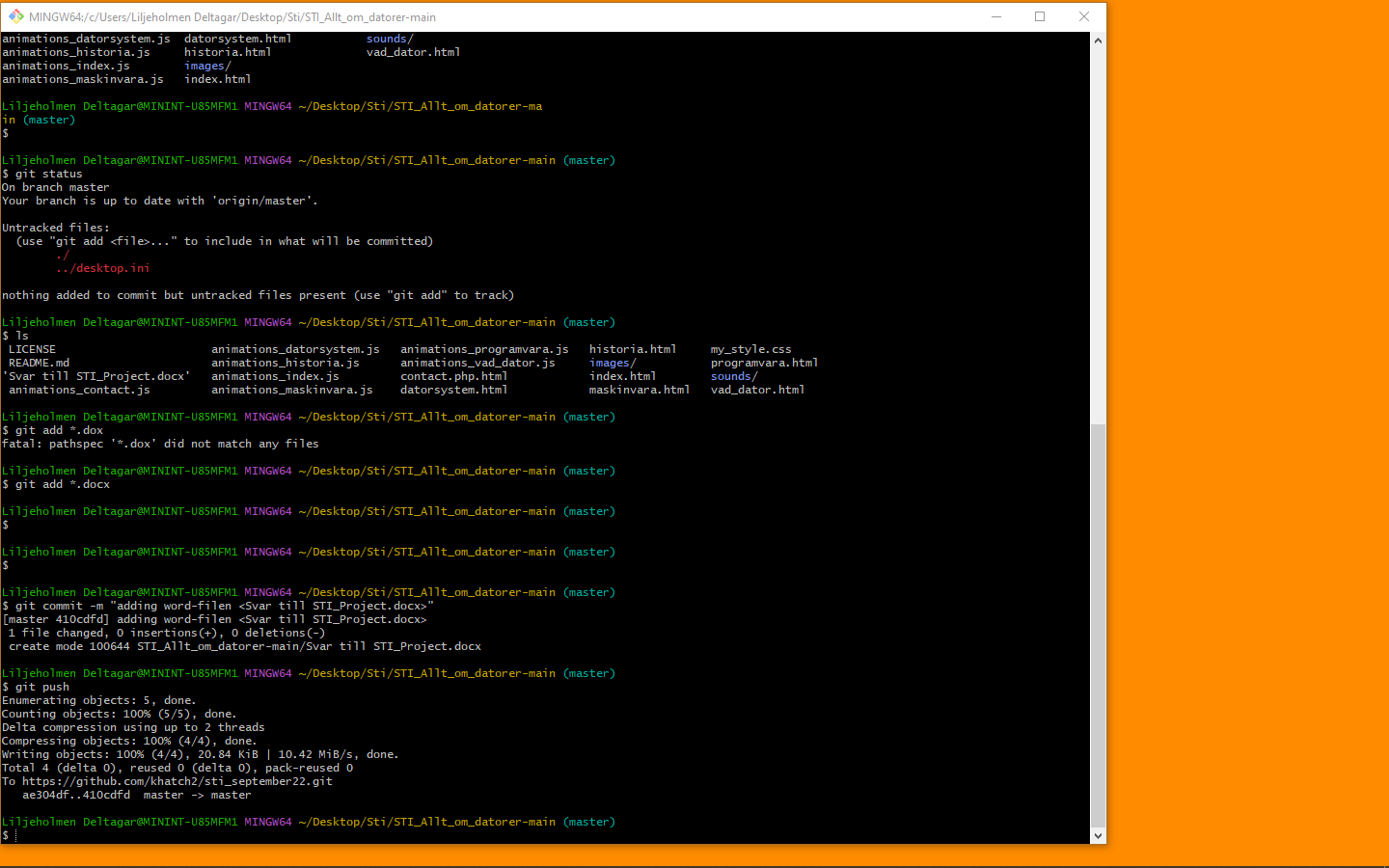
**Mina svar**







* Jag motiverar min versionshantering, d.v.s. här använde jag programmet “Git”, när jag behövdes att spara historian av versioner av min hemsidan “allt om datorer” och det har skett oftast vid varje behov, genom att addera och sedan kommenterar och till slut, kommitterar jag min ändringar till github.com.
* Här beskriver jag, och redogör jag skillnaderna och vad de följande sakerna används till:
  + Git som är ett executive-program som körs på dator för att:
    - Sedan kan man arbeta med filerna av sitt utvecklingsprojekt, och lagra versionshistorik av källkods filer och mappar..
    - Observera här att man genom programmet “Git” på sin egen dator kan arbeta också med “sourgeforge.net”, och inte bara med “github.com”.
  + GitHub som är: ett internet-bolag som erbjuder webbaserad och centraliserad lagring av versionshistorik för mjukvaru-utveckling, och i året 2018 e.Kr. förvärvades github av det stora bolaget Microsoft.
  + GitHub pages som är en tjänst från webbsiten Github.com, för att:
    - man kan publicera sitt/sina eget/egna repository/-ies som en webbsida.
* Sedan identifierar jag olika hybridlösningar för att skapa plattformsöverskridande applikationer. Vad används?
* Sedan redogör jag skillnaderna mellan web, native och hybridlösningar, på att:
  + Web-app är webbsida som gjords med HTML, CSS och JavaScript.
    - Men nackdelen med web-app är att den kräver mer resurser, dessutom att den är inte vän med sökmotorn Google.
  + Native-app är en applikation som byggdes för en plattform av operating system på 100% unikt sätt.
    - T.ex. För Apple Iphone IOS använder man här programmeringsspråk kallas “Objective-C” och/eller “Swift”.
    - T.ex. för Många andra mobiltelefoners tillverkare vilka används Google Android på deras mobiltelefoner, använder man här programmeringsspråk “Java” och/eller “Kotlin”.
    - Men nackdelen här är att nativa applikationer för Microsoft Windows kan inte köras på Apple Macintosh.
  + Hybrid-app är annan former av kodmiljöer, till exempel webbsida som köras i native-skal och/eller översättas till native på en special plattform av operating system.
    - Här med “React Native” använder man ett programmeringsspråk kallas för “JavaScript”.
    - Men nackdelar är:
      * att de hybrida appar bryr sig inte och att unifierar samma utseende av dem, men alltid utnyttjar default grafier av GUI av varje operative systems plattform.
      * De är långsammare än native-appar.
      * De är dyrare än de standarda webbappar, och då bör man att jobba med de saker som kallas för wrapper..
      * De är beroende av tredje-parts-lösningar.
  + Sedan reflekterar jag kring när man skulle vilja välja en framför de andra, på grund att jag först bör att räkna budget av att utveckla källkod för en app för Android och en annan källkod för plattformen IOS, eller det räcker bara med att skriva en källkod på t.ex. Unity eller “React Native” som kommer att köras på både “Android” och “IOS” att mindre antal timmar och mindre budget av pengar för utvecklingen.
* Sedan beskriver jag HTML, CSS och JS, plus hur kan jag utveckla på deras användningsområden i samband med web och APP-utveckling:
  + HTML är från engelska “HyperText Markup language”, på svenska “ett märkspråk för hypertext”, och är ett programmeringsspråk bara för webbläsarna, t.ex. Mozilla Firefox, Apple Safari och Google Chrome.
    - <h1>Rubrik</h1>
    - <p>Här är ett stycke av text</p>
  + CSS är från engelska “Cascading Style Sheets”, på svenska “stilmall”, som är ett programmeringsspråk för att skapa, uppdatera och utforma webbsidor.
    - body {background-color: #eaecf0;}
    - p {font-family: “Times New Roman”}
  + JS, alltså Javascript, är ett programmeringsspråk som framför allt används t.ex. till att flytta elements och saker inuti webbsidor, men även i andra sammanhang.
    - <script> console.log(“Hejsan, programmerare!”); </script>
* Jag skrev hemsidan, som heter “Allt om dator” får handla om allt om datorer.
  + OBS: Att här följande jag vissa krav från skolan “STI”.
* Jag har använt relevanta semantiska taggar, inte bara “div”:s, för att strukturera min sida av “index.html”, bland annat:
  + “header”, “section” och “footer”.
  + <https://github.com/khatch2/STI_Allt_om_datorer/blob/main/index.html>

* Jag har använt flera av de vanligare HTML-taggarna som form, input, img, table, header, footer, a, med mera. T.ex.:
  + <link href="https://fonts.googleapis.com/css?family=Merienda+One" rel="stylesheet" />
  + <img id="logo\_id" class="logo" alt="Allting om datorer" src="./images/hw01.gif" />
  + <audio controls id='my\_audio'> <source src="./sounds/001.mp3"> </audio>
  + <https://github.com/khatch2/STI_Allt_om_datorer/blob/main/index.html>
* För CSS skulle jag redan haft stylat sidan simpelt, d.v.s. att det behövs inte vara något mästerverk i design, och samt hade jag använt klasser, id:n, element, pseudo-klasser och hade jag vart mer specifika ¨an bara en selector i minst ett fall (kedja flera), t.ex.:
  + @charset "UTF-8";
  + #copyleft { display: inline-block; transform: rotate(180deg); }
  + .startpage > section:last-of-type { margin-right: 0; }
  + .startpage > section { display: inline-block; width: 20%; margin-right: 6%; vertical-align: top; }
  + body { margin: auto; /\* Centrera sidan \*/ width: 860px; font-family: 'Helvetica Neue', 'Liberation Sans', Calibri, Arial, sans-serif; font-size: 100%; }
  + <https://github.com/khatch2/STI_Allt_om_datorer/blob/main/my_style.css>
* För JS, alltså javascript, skulle jag redan haft manipulerat något element på sidan, t.ex.:
  + function moveLogo() { var pos = 0; var logo = document.getElementById('logo\_id'); logo.style.width = '215px'; logo.style.height = '170px'; logo.style.background = 'red'; logo.style.position = 'absolute'; var time\_interval = setInterval(move, 10); function move(e){ if (pos >= 840) { clearInterval(time\_interval); } else { pos ++; logo.style.right = pos+'px'; }
  + <https://github.com/khatch2/STI_Allt_om_datorer/blob/main/animations_index.js>
* Detta kunde jag vara vid t.ex. form submission, t.ex.:
  + <https://github.com/khatch2/STI_Allt_om_datorer/blob/main/contact.php.html>
* Dessutom jag gjorde att använda loopar, variabler och funktioner inom Javascript, t.ex.:
  + function move(e){ if (pos >= 215) { clearInterval(time\_interval); canvas.style.scale = '3'; } else { pos ++; write\_img.style.right = pos+'px'; databas\_img.style.right = 5\*pos+'px'; moveCanvas(3.1\*pos); }
  + <https://github.com/khatch2/STI_Allt_om_datorer/blob/main/animations_contact.js>

**Källor:**

* Khatchik Shahinyan, *Allt om datorer*, soucecode att: <https://github.com/khatch2/STI_Allt_om_datorer>
* Khatchik Shahinyan, *Allt om datorer*, Github pages att: <https://khatch2.github.io/STI_Allt_om_datorer/>

==========================

Students namn: Khatchik Shahinyan

=====================================================================