For å hente ut noe fra GitHub bruker vi git clone kommandoen. Denne kommandoen kloner alt fra gitt repositorien til din pc. Dette er viktig for oss siden som tidligere sagt bygger vi prosjektet vårt på en skjellet laget av en tidligere student. Denne skjellete og alt annet som vi vil ha fra GitHib kan vi hente ut med bruk av *git clone «link»*.

I figuren under har gruppen brukt git clone til å klone skjelettet.

Et bilde som inneholder tekst

Automatisk generert beskrivelse

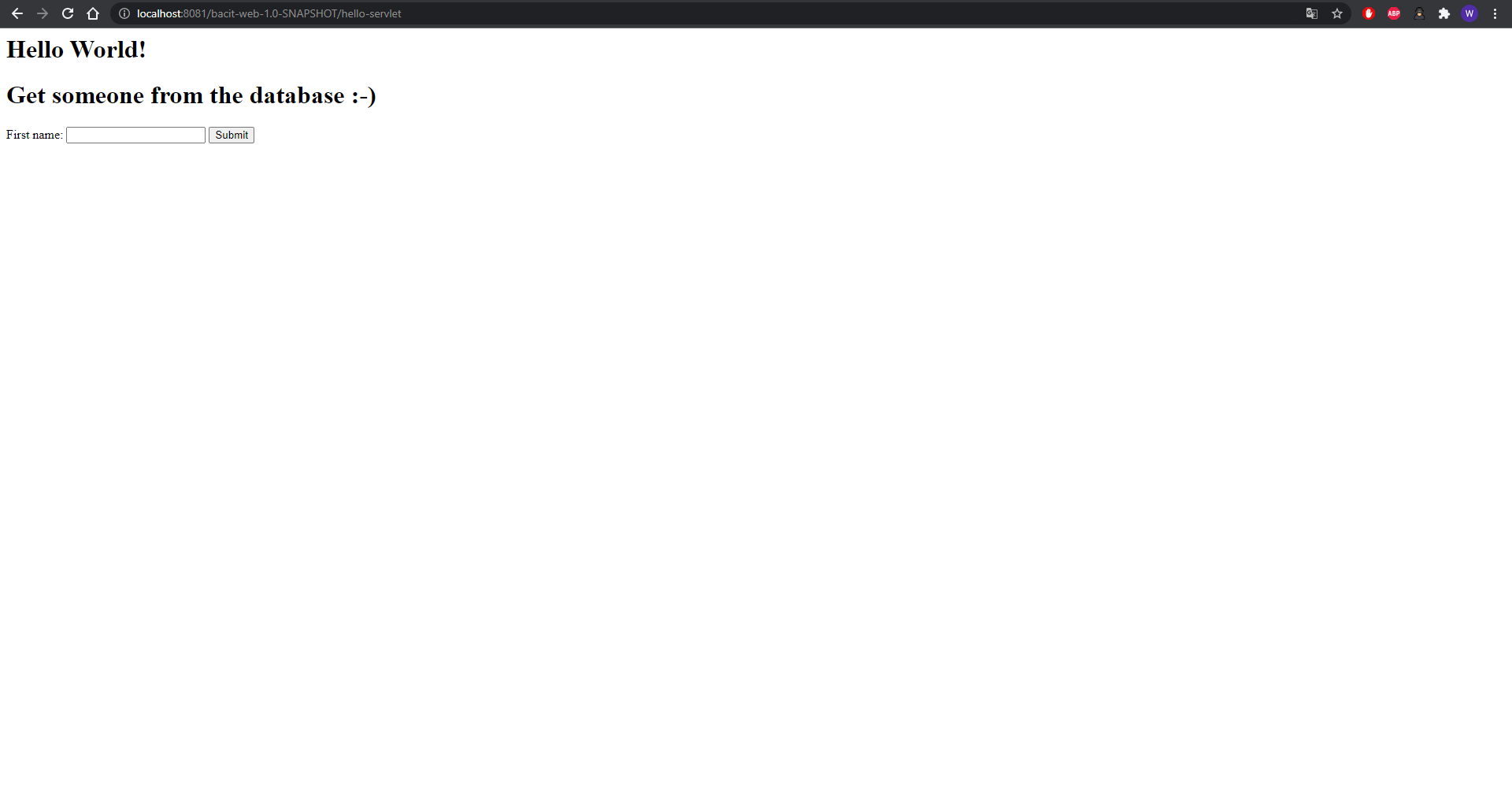
Et bilde som inneholder tekst

Automatisk generert beskrivelse

Etter å ha klonet repositorien fikk vi 2 filer (build.cmd(sh for Linux) og startDb.cmd). Disse filer inneholdte det kommandoene som skulle brukes til å bygge og kjøre skjelettet. For å bygge skjelettet kjørte vi build.cmd med ./build.cmd, og ./startDb.cmd for å starte databasen. Et bilde som inneholder tekst

Automatisk generert beskrivelse

Etter å ha kjørt kommandoen ./build.cmd fikk vi en link som kunne brukes til å se om vi har gjort alt riktig og samhandle med skjelettet og servleten. Men for å samhandle med databasen må startDb.cmd kjøre først.

Et bilde som inneholder tekst

Automatisk generert beskrivelse

Servlets:

En servlet er en simpel klasse som responderer til en bestemt type nettverksforespørsel og den vanligst er http. Den letteste måten å lage en servlet på er å lage en subklasse av javax.servlet.http.HttpServlet. Servlets kjører i noe som kalles for servlet containers(beholder), og denne «beholderen» håndterer nettverk siden som bl.a. tilkobling. I vårt prosjekt tar vi i bruk tomcat som er den mest vanlige servlet containeren.

For vårt prosjekt er det å lage og forstå servlets essensielt, siden vi kommer til å bruke det mye. Noen av av oppgavene som vi skal bruke servlets på er for eksempel, å hente ut informasjon fra databasen og legge noe til databasen.

HelloServlet og GetUserServlet er de to servletene som kommer med skjelettet som prosjektet vårt bygger seg på. GetUserServlet brukes til å hente ut informasjon om en bruker i databasen. I figuren under ser vi at den er i pakken bacit.web.bacit\_web og er en subklasse av HttpServlet. Den har en doGet metode som brukes til å hente ut informasjon om gitt «uname» med hjelp av objektet «model» som videre blir lagret i variabelen uname. Hvis den fungerer printer systemet ut fornavnet til gitt uname, hvis det oppstår en feil printer den feilen.

Et bilde som inneholder tekst

Automatisk generert beskrivelse

«model» er et objekt av klassen UserModel. I figuren under finnes det en getUser metode som benytter seg også av UserModel klassen. Denne metoden sjekker først om det er en tilkobling mellom databasen og servleten. Hvis det er ikke en kobling skriver den ut en feilmelding.

I metoden finnes det også en query som skal velge fra tabellen en rad hvor User\_firstName er lik «uname» og videre lagre alt informasjonen som fornavn etternavn fra den «uname» raden i objektet model og til slutt returnere model objektet.

Et bilde som inneholder tekst

Automatisk generert beskrivelse

Dbutils: