Sikkerhetskrav

Funksjonelle:

Systemet skal på alle input-felter ha inputvalidering og bruke prepared statements.
Fil opplastning skal bare tillate .png og .jpg filer ved opplasting, samtidig som at det skal være satt en viss maks størrelse på filen.
Systemet skal ha kryptering i form av HTTPS mellom klient og server.
Systemet skal, i tillegg til å ha en root bruker, ha andre brukere med egen definerte rettigheter avhengig av hvilke handlinger brukeren skal få lov til å utføre.
Systemet skal logge sikkerhetsrelaterte hendelser, slik som for eksempel brukerinnlogginger som feiler eller ved uvanlig høy trafikk (DoS).
Systemet skal kunne spore handlinger som blir gjort, hvem som utførte dem og når de skjedde.

Ikke funksjonelle:

Det skal være mulig å lese data uten å være innlogget (gjestebruker).
Systemet skal være enkelt for brukere med et mål - å få meldinger fra deg til dem.
Systemet må være raskt og kunne håndtere store mengder nåværende brukere samtidig.
Systemet skal ha så mye oppetid som mulig, bortsett fra nedetid grunnet vedlikehold.
Systemet skal følge GDPR's lover, å sikre brukerdata og lagring. Samtaler mellom student og forleser må dermed holdes privat og ikke komme på avveie.

Abuse cases

En angriper legger inn et script som kommentar i et inputfelt som deretter blir kjørt på siden.
En angriper benytter SQL-spørringer for å få tilgang til data som brukeren ikke skal ha tilgang til.
Angriper kommer seg rundt filvalidering og laster i stedet opp skadelig kode.
En angriper får tilgang til en annen brukers innlogget sesjon, og kan dermed utgi seg for dem inntil den blir stengt.
En angriper utnytter manglende begrensninger på antall forespørsler og fyller opp databasen med skrot-data.
En angriper endrer exif data på et bilde slik at det inneholder php kode som kan bli kjørt.
En angriper oppnår å kunne lese av php filer i systemet og kan dermed se data som login etc. til databasen.
En Angriper benytter SQL-spørringer for å få tilgang til data som brukeren ikke skal ha tilgang til.

En angriper får tilgang til root brukeren og får alle rettigheter.

En angriper finner git repositoriet og ser deretter dens informasjon.

En angriper finner et sikkerhetshull som gir dem tilgang til systemet gjennom en bakdør.

En angriper kan se data som blir sendt gjennom nettsiden ved hjelp av programmer som wireshark siden koblingen er HTTP.