



05.12.2025



REFLEKSJONSRAPPORT

IBE160 - Programmering med KI
AI Escape

M.E.Ds

Mathilde Granden Julseth - 201126
mathilde.g.julseth@himolde.no

Emilie Stenberg Silnes - 200870
emilie.s.silnes@himolde.no

Daniel Langnes Cric - 230950
daniel.l.cric@himolde.no

Mattias Ljøkjell - 231590
mattias.ljokjell@himolde.no

Innholdsfortegnelse

Innholdsfortegnelse.....	2
Utviklingsprosessen.....	3
Oversikt over prosjektet.....	3
Arbeidsmetodikk.....	3
Teknologi og verktøy.....	3
Utviklingsfaser.....	4
Fase 1 - Planlegging.....	4
Fase 2 - Utvikling.....	4
Utfordringer og løsninger.....	4
Tekniske utfordringer.....	4
Utfordring 1: Feil i filsystemet som stoppet commit og ødela story-context.....	4
Utfordring 2: Konflikter i kodeendringer når KI genererte oppdateringer.....	5
Utfordring 3: Ødelagt funksjonalitet etter KI-generert bildejustering.....	5
Samarbeidsutfordringer.....	6
KI-spesifikke utfordringer.....	6
Kritisk vurdering av KI sin påvirkning.....	7
Fordeler med KI-assistanse.....	7
Begrensninger og ulemper.....	7
Avhengighet og forståelse.....	7
Kreativitet og problemløsning.....	8
Sammenligning: Med og uten KI.....	8
Samlet vurdering.....	8
Etiske implikasjoner.....	9
Ansvar og eierskap.....	9
Transparens.....	9
Påvirkning på læring og kompetanse.....	10
Arbeidsmarkedet.....	10
Datasikkerhet og personvern.....	10
Teknologiske implikasjoner.....	11
Kodekvalitet og vedlikehold.....	11
Standarder og beste praksis.....	12
Fremtidig utvikling.....	12
Konklusjon og læring.....	12
Viktigste lærdommer.....	12
Hva ville dere gjort annerledes?.....	13
Anbefalinger.....	14
Personlig refleksjon (individuelt).....	14
Mathilde Granden Julseth.....	14
Emilie Stenberg Silnes.....	14
Daniel Langnes Cric.....	15
Mattias Ljøkjell.....	15

Refleksjonsrapport

Utviklingsprosessen

Oversikt over prosjektet

I dette prosjektet har vi utviklet en KI-drevet, tekstbasert escape room-applikasjon. Målet vårt var å lage et digitalt escape room hvor hvem som helst kunne spille uten å måtte være fysisk til stede. I stedet velger spilleren kommandoer for å klare å komme seg ut av rommet.

Hovedfokuset var å utvikle et interaktivt og intelligent spill der spilleren kommuniserer gjennom kommandoer som *look*, *use*, *take* og *read*, mens KI-en genererer kontekstavhengige beskrivelser, ledetråder og hint underveis.

Prosjektet demonstrerer hvordan KI kan fungere både som en kreativ historieforteller og som et pedagogisk verktøy, ved å støtte logisk tenkning, problemløsning og refleksjon hos spilleren.

Arbeidsmetodikk

Når det kom til arbeidsmetodikken vår, så samarbeidet vi på alt. Vi vurderte å dele opp oppgaven slik at folk kunne jobbe i hver sin branch, men vi tenkte heller at det var bedre å gjøre alt sammen, slik at det ikke ble noe surr eller usikkerhet. Vi følte også at dette ville minske sjansen for at problemer skulle oppstå.

Oppgavefordelingen falt inn naturlig underveis. Git fungerte som repositoriet vårt, hvor alle filene ble lagret og delt. Mathilde er god på å skrive presise og tydelige promter, så hun var ofte den som jobbet direkte i Visual Studio Code, mens resten av gruppen fulgte med via skjermdeling på Discord og bidro med forslag og vurderinger. Emilie skrev ned notater underveis for å gjøre det enklere for seg selv, Mattias og Daniel å starte på rapporten i Google Docs når fristen nærmet seg. På grunn av varselproblemer ble ikke Teams brukt mye, men det ble brukt for å diskutere med forelesere hvis vi trengte hjelp med noe.

Teknologi og verktøy

I dette prosjektet har vi brukt flere teknologier og verktøy for å utvikle et AI-basert, tekstbasert escape room-spill. Valgene er gjort med fokus på brukervennlighet, funksjonalitet og effektiv utvikling.

Frontend: Vi brukte **HTML**, **CSS** og **Tailwind**. Det gjorde det mulig å lage en rask og responsiv løsning med god struktur i koden, mens HTML og CSS ble brukt til oppbygging og design av spillet.

Backend: Backend-løsningen ble utviklet med **Python Flask**. Flask er et lett og fleksibelt rammeverk som gjorde det mulig å håndtere forespørsler, sende og motta data fra frontend og kommunisere med KI-tjenester og databasen på en effektiv måte.

Database: Vi brukte **Supabase**. Supabase ble brukt til lagring av data og håndtering av innlogging og autentisering.

KI-verktøy: I utviklingsarbeidet brukte vi **Gemini CLI, ChatGPT og Nano Banana pro** til blant annet koding, feilsøking, forklaringer, ideutvikling, KI-basert bildegenerator og bilderedigeringsmodell.

Andre verktøy: Til programmering og utvikling brukte vi **Visual Studio Code (VS Code)** som kodeeditor.

Utviklingsfaser

Fase 1 - Planlegging

Da vi begynte med planleggingen så begynte vi med å brainstorme med masse forskjellige ideer. Vi leste gjennom forslagene læreren ga oss. Vi først vurderte å lage en form for kostholdsapp, men ettersom mer tenking og noen som kom på en idé om å lage et slags spill, endret vi planen. Vi valgte å lage en tekstbasert KI-escape room, noe som fanget blikket vårt ganske fort.

Fase 2 - Utvikling

Da vi begynte med prosjektet, brainstormet vi om hva vi ville ha med, hvordan det kunne se ut. Vi ville prøve å visualisere det slik at det ville være lettere når det kom til å designe det og legge til funksjoner vi ville ha med.

Utfordringer og løsninger

Tekniske utfordringer

Utfordring 1: Feil i filsystemet som stoppet commit og ødela story-context

- **Problem:**

Da vi skulle committe og pushe endringer, oppdaget både vi og Gemini at det var noe galt i filsystemet. En viktig fil hadde blitt opprettet i bilde mappen. Dette førte til at Gemini stoppet commit-prosessen fordi filpllasseringen ikke samsvarer med forventet struktur.

Da Gemini ble bedt om å flytte filen tilbake dit den skulle være, oppsto et nytt problem: Create Story Context-steget ble nullstilt og filene slettet, så developer agenten fikk ikke noe å jobbe med, selv om alle storyene var validert som "Ready for dev".

- **Løsning:**

For å komme videre måtte vi involvere Scrum Master agenten for å få prosessen tilbake på sporet. Som nevnt ba vi Gemini flytte filen til korrekt mappe, men det kom ikke opp som et alternativ. Da flyttet vi den manuelt uten problem, men vi måtte begynne med hele create-story-context prosessen på nytt, og den kjørte alle storyene inkludert validering, noe som tok mange timer.

- **KI sin rolle:**

I denne situasjonen skapte KI-verktøyet problemet. Det var KI som opprettet filen i feil mappe, og greide ikke å rette opp feilen da vi ba den om det. I stedet gjorde Gemini situasjonen verre ved å nullstille og slette det validerete Create Story Context-steget. Dette stoppet arbeidsflyten fullstendig og gjorde at vi måtte starte en tidkrevende prosess på nytt. Erfaringen viste at KI-verktøyet kan utføre uforutsigbare handlinger som skaper ekstra arbeid, og at man ikke alltid kan stole på at det håndterer filsystem eller prosessflyt på en stabil måte.

Utfordring 2: Konflikter i kodeendringer når KI genererte oppdateringer

- **Problem:**

Etter å ha startet med UX-design, gjorde vi små justeringer underveis, og skulle på et tidspunkt endre litt på en farge, og da fikk vi beskjed om at contextvinduet var brukt opp - og Gemini kunne ikke fullføre handlingen sin. Vi ville da commite endringene for å starte på nytt, men det lot seg ikke fullføre fordi KI-en måtte fullføre sine oppgaver først.

- **Løsning:**

Problemet løste seg ved å godkjenne at Gemini fikk fullføre den avbrutte handlingen knyttet til fargeendringen. Da KI-en gjennomførte sin prosess, ble fargene endret som ønsket, og vi kunne deretter committe og pushe uten at feilmeldinger dukket opp igjen.

- **KI sin rolle:**

Her var KI-en både en hjelp og en hindring. På den ene siden kunne den gjøre raske endringer for oss, men på den andre siden skapte dens automatiske prosesser konflikter i arbeidsflyten vår.

Utfordring 3: Ødelagt funksjonalitet etter KI-generert bildejustering

- **Problem:**

Etter å ha lånt *Nano Banana Pro* av foreleser Bård ønsket vi at Gemini skulle tilpasse bildene slik at de ikke ble kuttet på midten og passet bedre inn i den definerte bildeboksen i designet. Resultatet så identisk ut som før, men med unntak av at knappene sluttet å fungere. KI-en hadde sannsynligvis påvirket deler av layout- eller komponentstrukturen.

- **Løsning:**

For å løse problemet valgte vi å bruke ChatGPT til å generere en forlenget versjon av bildet. Deretter redigerte vi bildet manuelt for å tilpasse det til bildeboksen, og lastet det nye bildet opp i prosjektets bildemappe. Samtidig forklarte vi til Gemini at knappene hadde sluttet å fungere etter bildejusteringen. Da vi presiserte dette, rettet KI-en umiddelbart opp i layoutfeilene og gjenopprettet funksjonaliteten i komponentene.

- **KI sin rolle:**

KI-verktøyene spilte en todelt rolle i denne utfordringen. På den ene siden skapte Gemini uforutsigbare endringer som førte til at fungerende komponenter ble ødelagt. På den andre siden bidro både ChatGPT og Gemini til å løse problemet: ChatGPT hjalp oss med bildebehandlingen, og Gemini korrigerte den feilaktige komponentstrukturen etter at vi gjorde problemet tydelig for den. Erfaringen viste at KI kan være nyttig i kreative og tekniske prosesser, men at den også kan introdusere feil som krever manuell kontroll og tydelig kommunikasjon.

Samarbeidsutfordringer

Vi opplevde få samarbeidsproblemer i prosjektet. Den eneste utfordringen vi støtte på i starten var knyttet til kommunikasjon på Teams. Noen av gruppemedlemmene hadde varslingsproblemer, noe som førte til at beskjeder ikke alltid ble fanget opp. Dette skapte litt surr rundt koordinering de første dagene, men vi fant fort ut at Discord fungerte best for alle.

Gruppen samarbeidet godt gjennom hele prosjektet: vi diskuterte, stilte spørsmål, og var enige om beslutninger underveis. Det har ikke oppstått konflikter eller misforståelser som påvirket arbeidsflyten, og vi opplevde kommunikasjonen som både tydelig og konstruktiv.

KI-spesifikke utfordringer

Bruken av kunstig intelligens i prosjektet var nyttig, men førte også med seg flere utfordringer. En av utfordringene var at KI-verktøyene til tider genererte feil eller ufullstendig kode. Enkelte løsningsforslag inneholdt logiske feil eller var ikke godt nok tilpasset prosjektets behov.

En annen utfordring var misforståelser i kommunikasjonen med KI. Dersom spørsmålene våre ikke var presise nok, kunne KI tolke oppgaven på en annen måte enn det vi ønsket, noe som førte til svar som ikke løste problemet fullt ut. Noen ganger fikk vi svært gode og direkte brukbare forslag, mens vi andre ganger måtte gjøre flere manuelle endringer for å få koden til å fungere riktig.

I tillegg opplevde vi at det av og til tok lang tid å få svar fra KI-verktøyet. Dette kunne bremse arbeidsflyten, spesielt når vi satt fast i problemer og var avhengige av raske avklaringer for å komme videre i arbeidet.

For å håndtere disse utfordringene innførte vi grundig testing og manuell kvalitetssikring av alle kode som ble foreslått av KI. Vi ble også flinkere til å stille mer presise og detaljerte spørsmål, og brukte i tillegg dokumentasjon, egne vurderinger og samarbeid i gruppen for å sikre at løsningene var riktige. Når det tok lang tid å få svar fra KI, fortsatte vi å jobbe videre med andre deler av prosjektet for å utnytte tiden best mulig.

Kritisk vurdering av KI sin påvirkning

Fordeler med KI-assistanse

KI-assistanse har sine fordeler. Arbeidet gikk raskere, spesielt problemløsing som debugging, selve strukturen av koden og generering av forslag til løsninger. Denne bistanden gjorde at vi kunne jobbe mer effektivt. Samtidig bidro KI-verktøyet til økt læring fordi den forklarte konsepter, og viste oss prosessene den fullførte, samt at den kom med alternative løsningsmåter. Kvaliteten på koden ble da bedre ved hjelp av kontinuerlig optimalisering, med en ryddig og oversiktlig syntaks. Totalt sett fikk sluttresultatet et profesjonelt og forståelig uttrykk.

Begrensninger og ulemper

Selv om kunstig intelligens har vært et nyttig hjelpemiddel i prosjektet, har vi også opplevd flere begrensninger og ulemper ved bruken. En av de største utfordringene var at KI til tider ga feil eller dårlige løsninger. Noen ganger fungerte ikke koden slik den skulle, eller den var ikke tilpasset hvordan prosjektet vårt faktisk var bygget opp. Vi opplevde også at noen svar var overfladiske eller manglet viktige detaljer.

Feilene ble som oftest oppdaget gjennom testing, feilmeldinger i konsollen og at funksjoner ikke oppførte seg som forventet. For å håndtere dette måtte vi gå gjennom koden manuelt, bruke dokumentasjon og samarbeide i gruppa for å rette feilene. Dette lærte oss at KI ikke alltid er pålitelig, og at menneskelig kontroll er helt nødvendig.

Avhengighet og forståelse

Det var perioder hvor vi kunne bli litt for avhengige av KI, spesielt når vi stod fast i problemer. I stedet for å prøve selv først, kunne det være fristende å spørre KI med en gang. Dette kunne i noen tilfeller gjøre at vi lærte mindre på egen hånd, fordi vi fikk ferdige løsninger i stedet for å finne dem selv. Samtidig oppdaget vi etter hvert at vi måtte forstå løsningene selv for å kunne bruke dem videre. Når vi begynte å stille mer presise spørsmål og faktisk sette oss inn i svarene, ble KI mer et støtteverktøy enn en erstatning for egen kunnskap.

Kreativitet og problemløsning

KI påvirket også kreativiteten vår på både positive og negative måter. På den ene siden ga KI mange ideer og forslag som kunne være inspirerende. På den andre siden merket vi at vi noen ganger fulgte KI sine forslag litt for tett, i stedet for å utvikle helt egne løsninger. Det var også situasjoner hvor KI foreslo standardløsninger som kunne begrense kreativ tenkning, for eksempel ved oppbygning av spillet eller løsning av tekniske problemer. I slike tilfeller måtte vi bevisst velge å tenke selv og prøve egne ideer før vi eventuelt brukte KI som støtte.

Sammenligning: Med og uten KI

Dersom prosjektet hadde blitt gjennomført uten bruk av kunstig intelligens, ville arbeidsprosessen vært betydelig annerledes. Uten KI som støtteverktøy måtte vi i større grad ha basert oss på egen kunnskap, lærebøker, dokumentasjon og nettsøk for å finne løsninger på problemer som oppstod underveis. Dette ville føre til lengre tidsbruk, spesielt i perioder hvor vi stod fast i utviklingsarbeidet. Vi ville også ha delt gruppen mer opp, eller prosjektet opp i flere deler. Alle ville ha fått hver sin del å gjøre.

KI bidro ofte med raske forklaringer og eksempler som gjorde det lettere å komme videre. Samtidig kan det sies at enkelte deler også kunne vært lettere uten KI, fordi vi da hadde vært tvunget til å finne løsninger selv hele veien, noe som potensielt kunne gi enda dypere forståelse for alle deler av koden.

Når det gjelder sluttresultatet, er det sannsynlig at prosjektet med KI hadde blitt enklere og mindre omfattende. Bruken av KI gjorde det mulig å teste flere løsninger, rette feil raskere og implementere funksjoner vi ellers kanskje ikke hadde hatt tid eller kompetanse til å utvikle. Samtidig er det mulig at kvaliteten på enkelte deler av koden kunne vært mer gjennomarbeidet dersom alt var skrevet manuelt fra bunnen av.

Alt i alt har KI bidratt til en mer effektiv utviklingsprosess, raskere problemløsning og et mer avansert sluttprodukt. Uten KI ville prosjektet sannsynligvis fortsatt vært gjennomførbart, men både prosess og resultat ville vært tydelig mer begrenset.

Samlet vurdering

Kunstig intelligens har totalt sett vært en netto positiv faktor for sluttresultatet i prosjektet. KI bidro til økt effektivitet, raskere feilsøking og støtte i utviklingsprosessen, noe som gjorde det mulig å utvikle et mer avansert og funksjonelt AI Escape Room enn det vi sannsynligvis ville klart uten KI.

For oss var det likevel slik at KI noen ganger var treg å få svar fra, mens den andre ganger ga raske og nytte svar. Dette gjorde at KI både kunne være et effektivt hjelpemiddel og en faktor som til tider bremset arbeidsflyten. Den viktigste

lærdommen for oss var at KI er et effektivt og tidsbesparende hjelpemiddel i utviklingsprosessen, men at det samtidig er helt avgjørende at vi selv forstår, tester og kvalitetssikrer løsningene før de tas i bruk.

Etiske implikasjoner

Ansvar og eierskap

Når KI brukes i utviklingsarbeid oppstår det spørsmål om hvem som faktisk har ansvaret for koden. Selv om KI kan generere store deler av løsningen, er det utviklerne som har det endelige ansvaret for å forstå, kontrollere og godkjenne koden som tas i bruk.

Kvalitetssikring blir dermed en sentral del av arbeidet siden man ikke kan forutsette at KI gir optimale eller korrekte løsninger. Alle koder må gjennom manuell vurdering, gjennomgang og kontinuerlig testing for å sikre riktig funksjonalitet.

Når det gjelder opphavsrett og eierskap, skaper KI-generert kode nye utfordringer. Generelt regnes koden som "ikke-opp-havrettslig" fordi KI-systemer ikke kan stå som rettighetshavere. Eierskapet ligger derfor hos utviklerne og prosjektgruppen som bruker og tilpasser koden. Likevel bør man være oppmerksom på at KI-modeller kan være trent på store datamengder som inneholder eksisterende kode, noe som gjør det viktig å følge retningslinjer for åpen kildekode, dokumentere arbeidsprosessen og sikre at man ikke kopierer lisensiert materiale uten tillatelse.

Transparens

Det bør være tydelig og transparent når kunstig intelligens er brukt i et prosjekt. Åpenhet om KI-bruk handler om ærlighet, faglig integritet og tillit. Når man er åpen om hvordan KI har blitt brukt, blir det lettere for andre å vurdere arbeidet på en rettferdig måte og forstå hvordan resultatet faktisk er blitt til. KI sin rolle bør også dokumenteres på en tydelig måte. Dette kan for eksempel gjøres ved å beskrive hvilke KI-verktøy som er brukt, hva de er brukt til for eksempel feilsøking, kodeforslag, idéutvikling eller tekstproduksjon, og hvilke deler av prosjektet som i hovedsak er laget manuelt. Slik dokumentasjon gjør det klart hva som er personens eget arbeid, og hva som er støttet av KI.

Dersom man ikke er åpen om bruk av KI, kan det få flere konsekvenser. Det kan føre til mistillit, feil vurdering av kompetanse og i verste fall regnes som juks eller brudd på skolens regler. I tillegg kan man gå glipp av viktig refleksjon rundt egen læring dersom KI-bruken skjules. Å være åpen om KI-bruk bidrar derfor både til rettferdig vurdering, bedre læring og mer ansvarlig bruk av teknologien.

Påvirkning på læring og kompetanse

Bruken av kunstig intelligens kan ha stor påvirkning på hvordan man utvikler kunnskap og ferdigheter innen IT. Dersom man blir for avhengig av KI, kan det føre til at egen problemløsning og forståelse svekkes over tid. I stedet for å selv analysere feil og finne løsninger, kan man bli vant til å la KI gjøre jobben. Dette kan påvirke fremtidig kompetanse negativt dersom man ikke er bevisst på hvordan man bruker verktøyet. Ved høy KI-avhengighet risikerer man å ikke utvikle viktige ferdigheter som grunnleggende programmeringsforståelse, feilsøking, logisk tenkning og selvstendig problemløsning. Disse ferdighetene er helt sentrale for å kunne jobbe som utvikler på lang sikt. Dersom man kun kopierer løsninger, lærer man mindre om hvorfor ting fungerer.

Det er derfor viktig å finne en god balanse mellom effektivitet og læring. KI kan brukes som et effektivt støtteverktøy for forklaringer, ideutvikling og feilsøking, men det er viktig at man også prøver selv først og setter seg inn i løsningene som anbefales. Når KI brukes på en bevisst og kritisk måte, kan det både øke effektiviteten og styrke læringen samtidig.

Arbeidsmarkedet

Utbredt bruk av kunstig intelligens vil i stor grad påvirke fremtidige jobber innen IT. Mange rutineoppgaver som tidligere har vært gjort manuelt, vil i økende grad kunne automatiseres. Dette kan føre til at enkelte arbeidsoppgaver forsvinner eller endres, samtidig som det åpnes for nye typer stillinger innen utvikling, drift, sikkerhet og KI-teknologi. KI kan gjøre utviklingsprosesser mer effektive, men vil ikke erstatte behovet for menneskelig kompetanse.

Noen roller vil bli mindre viktige, for eksempel utviklere som hovedsakelig jobber med enkle og repeterende oppgaver, som ren kodeproduksjon uten større ansvar for struktur og logikk. Samtidig vil flere roller bli viktigere, som systemutviklere med god helhetsforståelse, KI-spesialister, dataingeniører, sikkerhetsekspertar og arkitekter som kan planlegge og kontrollere komplekse systemer. Evnen til å forstå, styre og kvalitetssikre KI vil bli stadig viktigere.

Når det gjelder egne refleksjoner rundt fremtidig karriere i en KI-drevet verden, ser vi både muligheter og utfordringer. KI kan være et kraftig hjelpemiddel som gjør oss mer effektive og gir rom for mer kreativt og avansert arbeid. Samtidig blir det enda viktigere å utvikle egne ferdigheter og forstå teknologien i dybden, slik at vi ikke blir avhengige av KI, men kan bruke det som et verktøy. Vi tror at utviklere som er villige til å lære nytt og tilpasse seg, vil ha gode jobbmuligheter også i fremtiden.

Datasikkerhet og personvern

I dette prosjektet delte vi i hovedsak kode, feilmeldinger og beskrivelser av funksjonalitet med KI-verktøy som ChatGPT og Gemini CLI. I tillegg ble

e-postadresser brukt i forbindelse med innlogging og bruk av Gemini og ChatGPT. Utover dette delte vi ikke sensitiv informasjon som passord eller andre personopplysninger. Dette var et bevisst valg for å ivareta datasikkerhet og personvern.

Det finnes likevel flere potensielle risikoer ved å dele kode og data med KI. Deling av personopplysninger, som e-post, kan føre til uønsket bruk av informasjonen dersom sikkerheten hos tjenesteleverandøren blir brutt. Deling av kode kan også innebære risiko for at sårbarheter blir eksponert, eller at prosjektinformasjon kommer på avveie. I tillegg er det ikke alltid helt klart hvordan data lagres og behandles av KI-tjenester.

Når man bruker KI i utviklingsarbeid, er det derfor svært viktig å tenke sikkerhet hele tiden. Man bør kun dele det som er absolutt nødvendig, aldri dele passord eller sensitive opplysninger, og alltid lese personvernerklæringer og vilkår for tjenestene man bruker. All KI-generert kode bør også gjennomgås manuelt for å sikre at den ikke inneholder sikkerhetshull. Bevissthet rundt sikkerhet og ansvarlig bruk av KI er avgjørende for trygg bruk av teknologien.

Teknologiske implikasjoner

Kodekvalitet og vedlikehold

Bruken av KI-generert kode kan påvirke langsiktig vedlikehold av et system på både positive og negative måter. På den positive siden kan KI bidra til rask utvikling, færre skrивfeil og forslag til effektive løsninger. Samtidig kan det oppstå utfordringer dersom koden ikke er godt nok tilpasset prosjektets struktur eller dersom utviklerne ikke fullt ut forstår hvordan løsningen fungerer. Dette kan gjøre videreutvikling og feilretting mer krevende over tid.

KI-generert kode er ikke alltid like forståelig som menneskeskrevet kode. Selv om koden ofte fungerer teknisk, kan den mangle tydelige kommentarer, ha uklar struktur eller bruke løsninger som ikke er intuitive for utviklere som senere skal jobbe med prosjektet. Menneskeskrevet kode er ofte bedre tilpasset prosjektets behov og enklere å lese, spesielt når utvikleren selv har full kontroll over logikken bak løsningene.

En annen utfordring er debugging av KI-generert kode. Når man ikke selv har skrevet koden fra bunnen av, kan det være vanskeligere å forstå hvorfor en feil oppstår. Feil kan også være skjult i komplekse løsninger som KI har foreslått. Dette gjør det ekstra viktig med grundig testing, god dokumentasjon og manuell gjennomgang av koden før den tas i bruk. Alt i alt kan KI være et nyttig verktøy for utvikling, men for å sikre god kodekvalitet og enkelt vedlikehold på lang sikt, må utviklere ha god forståelse for koden som brukes og ikke stole blindt på KI-genererte løsninger.

Standarder og beste praksis

KI følger ikke alltid etablerte industristandarder eller beste praksis. Selv om forslagene ofte fungerer teknisk, kan de noen ganger være utdaterte, unødvendig komplekse eller lite i tråd med moderne utviklingsmetoder. Dette kan for eksempel være bruk av gamle syntakser, unødvendig komplekse funksjoner eller mønstre som ikke passer inn i strukturen til prosjektet.

Derfor er det viktig å alltid validere KI-forslag mot dokumentasjon og egne faglige vurderinger. KI kan være et effektivt støtteverktøy, men kvaliteten sikres først når utvikleren selv vurderer om løsningen faktisk følger gjeldende standarder.

Fremtidig utvikling

Kunstig intelligens vil i stadig større grad påvirke hvordan programvare utvikles i fremtiden. KI vil bli et enda mer integrert verktøy i hele utviklingsprosessen, fra planlegging og design til koding, testing og feilsøking. Mange rutineoppgaver vil i større grad kunne automatiseres, noe som kan føre til raskere utvikling, færre enkle feil og mer effektive arbeidsprosesser. Samtidig vil kravene til kvalitet, sikkerhet og etisk bruk av teknologi bli enda viktigere.

Fremover vil det være spesielt viktig for utviklere å ha god grunnleggende programmeringsforståelse, slik at de kan vurdere og kontrollere det KI foreslår. I tillegg vil ferdigheter innen problemløsning, kildekritikk og feilsøking bli stadig viktigere, siden utviklere må kunne vurdere om KI-genererte løsninger faktisk er riktige og sikre. Også forståelse for datasikkerhet, personvern og ansvarlig bruk av KI vil være sentrale ferdigheter i fremtiden.

Anbefaling er at KI bør brukes som et støtteverktøy – ikke en erstatning for egen kompetanse. KI egner seg godt til idéutvikling, forklaringer, forslag til kode og feilsøking, men alt innhold bør alltid kontrolleres av utvikleren selv. Det er også viktig å bruke KI bevisst, stille presise spørsmål og kombinere svarene med dokumentasjon og egne vurderinger. Dersom KI brukes på en ansvarlig måte, kan det bli et svært nyttig hjelpemiddel i framtidig programvareutvikling.

Konklusjon og læring

Viktigste lærdommer

Tålmodighet i utviklingsprosessen

- Vi erfarte at det å jobbe med KI og kode krever tilmodighet, spesielt når løsninger ikke alltid fungerer. Flere ganger måtte vi prøve på nytt og ta et steg tilbake før ting fungerte slik vi ønsket. Dette lærte oss å holde roen og jobbe systematisk framfor å stresse etter raske svar.

Struktur, nøyaktighet og god kommunikasjon

- Prosjektet viste hvor viktig det er å jobbe strukturer og kommunisere tydelig i gruppa, med forelesere og KI-verktøyet. Ved å jobbe så mye sammen som vi gjorde, unngikk vi misforståelser og kunne jobbe effektivt. Nøyaktigheten i både kode, prompts og dokumentasjon gjorde utviklingen enklere.

Forståelse for KI-ens arbeid og “tankemønster”

- Gjennom prosjektet lærte vi hvordan KI tenker og hvorfor den gir bestemte svar. Dette hjalp oss med å skrive bedre prompts, tolke forslagene riktig og vite når vi burde be om en annen løsning. Å forstå hvordan KI jobber gjorde oss også mer kritiske til svarene den ga.

Økt forståelse for kode og tekniske løsninger

- Selv om ikke alle hadde mye programmeringserfaring fra før, lærte vi mye av å lese gjennom og analysere koden KI skrev. Etter hvert ble det enklere å forstå strukturen, logikken og hvordan funksjoner henger sammen. Dette prosjektet gjorde at vi fikk en mye bedre grunnleggende kodeforståelse.

Viktigheten av validering og kritisk vurdering

- Validering av prosessene ble tidlig en favorittfunksjon hos oss, da det bekreftet eller avkreftet hva som var rett og galt. Selv om den til tider kunne ta lang tid, ble det for oss helt nødvendig for å få en god oversikt over hva som skulle forbedres, og hvor godt vi lå an i forhold til kriteriene.

Hva ville dere gjort annerledes?

En viktig faktor som påvirket prosjektet, var at vi fikk tilgang til verktøyene og nødvendig informasjon ganske sent i semesteret. Dette gjorde at vi startet senere enn planlagt, noe vi ikke hadde kontroll over. Hadde vi fått startet tidligere, ville vi kunnet bruke mer tid på hver fase, testet flere løsninger og jobbet mer forutsigbart.

Innen samarbeid og organisering ville vi helt klart valgt å møtes oftere og hatt faste møtetidspunkter. Det ville gitt bedre flyt i arbeidet og raskere avklaringer. Selv om samarbeidet fungerte godt, ser vi i etterkant at mer struktur kunne gjort prosessen enda smidigere.

Vi innser også at dersom vi hadde visst tidligere hvor mye tid som faktisk går med til å vente på Gemini, ville vi sannsynligvis møttes oftere i starten av prosjektet. Lange responstider gjorde progresjonen uforutsigbar - og bare mens vi skriver dette, har vi ventet 2 timer og 45 minutter på én prompt. Med denne kunnskapen i forkant kunne vi lagt opp arbeidet annerledes og jobbet mer effektivt sammen.

Alt i alt er vi fornøyde med resultatet og samarbeidet, men med tidligere oppstart og mer struktur i møtene kunne prosessen blitt langt mindre tidspresset.

Anbefalinger

For å få mest mulig ut av KI bør den brukes som et støtteverktøy, ikke som en erstatning for egen kunnskap. Still tydelige og presise spørsmål, og bruk KI både til forklaringer, idéutvikling og feilsøking, ikke bare til å få ferdig kode. Kombiner alltid svarene fra KI med dokumentasjon og egne vurderinger for å sikre gode løsninger.

Samtidig er det viktig å være bevisst på fallgruvene. En av de største feilene man kan gjøre er blind kopiering av kode uten å forstå hva den gjør. For stor avhengighet av KI kan føre til svakere læring og dårligere problemløsingsevner. Man må også være forsiktig med å dele sensitiv informasjon, som passord, eller personopplysninger, med KI-verktøy.

Personlig refleksjon (individuelt)

Mathilde Granden Julseth

Prosjektprosessen har for meg vært læringsrik, givende på mange måter, langdryg, morsom og litt irriterende. I løpet av prosjektet ble det jeg som jobbet mest med selve Gemini og VSCode, og det har utfordret meg på måter jeg ikke kunne ha forestilt meg. Man lærer å formulere setningene sine på en helt annen måte, og forståelsen min for generativ AI generelt er høyst utviklet, samt at jeg har fått testet grenser når det kommer til tålmodighet mot teknologi.

Tross dette må jeg si at det har vært både underholdende og fascinerende å se KI-en jobbe. Jeg innser dens styrker og dens uendelige potensial, og er overbevist om at dette er og blir et svært viktig verktøy for de fleste fagområder. Det har blitt mye venting, noe som ikke vanligvis er en veldig stor del av aktivt studiearbeid til vanlig - og det har vist seg å være vanskeligere enn trodd. Det er mange, mange timer arbeid uten å egentlig "ha gjort noe". Men til gjengjeld, har vi oppnådd et gigantisk prosjekt på kort tid. Jeg er glad for at det var et gruppeprosjekt, og at vi har fått erfare dette sammen.

Emilie Stenberg Silnes

Å programmere med KI er noe jeg ikke hadde vært borti før, så jeg gledet meg mye til å starte med prosjektet. Etter vi kom i gang med prosjektet så gikk det opp for meg at jeg har undervurdert hvor kraftig KI faktisk er. Gemini CLI kom ofte med gode forslag som ingen av oss i gruppa hadde tenkt på, og guidet oss gjennom flere situasjoner der vi satt fast.

Samtidig merket jeg at det å se gjennom koden som Gemini skrev, gjorde det enklere for meg å lære og forstå hvordan ting faktisk fungerer. Jeg kunne ikke mye koding fra før, men likevel ble det lettere og lettere å lese og henge med i koden etter hvert. Bare det å formulere tydelige setninger ble også enklere etter å ha brukt så mange presise prompter. Dette har vært en stor del av læringsutbyttet mitt.

Jeg har også lært mye om samarbeid siden vi har jobbet tett sammen, delt skjerm, diskutert og funnet ut av problemer som et team. Det gjorde at jeg aldri følte meg alene i prosessen, og at vi hele tiden kunne bygge videre på hverandres ideer. Dette prosjektet har vært stressende og noen ganger frustrerende, men alt i alt utrolig gøy!

Daniel Langnes Cirim

Min refleksjon på dette prosjektet er at det var et ganske artig prosjekt. På starten kom vi alle på masse forskjellige ideer på hva vi kunne lage. Det første jeg tenkte på var et escape room-lignende spill. Da kom vi på ideen om et AI tekstbasert escape room-spill, noe som fanget meg veldig. Gjennom hele prosjektet har gruppen jobbet bra sammen. Det eneste problemet vi hadde var for det meste at vi måtte finne tid som passet alle for å jobbe med prosjektet. Det å bruke Gemini eller AI for å lage en app var ganske artig og det var ganske behagelig å se at alt ble gjort for deg, men det var litt irriterende også. Det var noen ganger problemer hvor Gemini ikke funket, eller ting vi gjorde som ikke ble lagret. AI-en var ikke perfekt for å si det slikt, men ettersom vi ga mer detaljert info eller instruksjoner, så ble det lettere å bruke AI-en uten problemer. Alt i alt, så var jeg ganske fornøyd med dette prosjektet, og gleder meg veldig til at det blir ferdig, slik at vi kan få spilt det.

Mattias Ljøkjell

Arbeidet med AI Escape Room-prosjektet har vært både spennende, utfordrende og lærerikt. Allerede i startfasen fikk vi ideen om et escape room-lignende spill, og det var motiverende å se hvordan dette faktisk utviklet seg til et ferdig, tekstbasert spill med kunstig intelligens. Prosjektet har gitt meg en bedre forståelse av hvordan KI kan brukes, både som et hjelpemiddel i koding og som en aktiv del av selve spillopplevelsen. Teknologier som HTML, Tailwind og Python Flask kan kobles sammen i et fullstendig prosjekt. Samtidig har jeg blitt mer bevisst på hvor viktig struktur, testing og feilsøking er for at et prosjekt skal fungere slik det er ment. Samarbeidet i gruppa har gått veldig bra, selv om det til tider var utfordrende å finne tidspunkter som passet for alle i gruppa. Bruken av KI-verktøy som ChatGPT og Gemini har vært både nyttig og til tider frustrerende. Alt i alt er jeg veldig fornøyd med prosjektet. Det har gitt meg verdifull erfaring innen programmering, samarbeid og bruk av kunstig intelligens. Prosjektet har også gjort meg mer motivert for videre læring.