



Prosjektbeskrivelse, bachelorprosjekt 2026, “KI-basert system for analyse og strukturering av tilgjengelighetsfunn til GitHub Issues”

Gruppemedlemmer

Gruppemedlemmene	LinkedIn
Christoffer Staupe	https://www.linkedin.com/in/christoffer-staupe-700138381/
Erlend Falkeid	https://www.linkedin.com/in/erlend-falkeid-43ba91380/
Nina Huynh	https://www.linkedin.com/in/nina-huynh-92a0bb327/
Mahsa Rasoli	https://www.linkedin.com/in/mahsa-rasoli-494126293/
Tringa Konxheli	https://www.linkedin.com/in/tringa-konxheli-505451353/

1. Oppdragsgiver

Oppdragsgiver for prosjektet er Tingtun AS, et norsk selskap som arbeider med utvikling av teknologi og tjenester knyttet til universell utforming og tilgjengelighet i digitale løsninger. Selskapet har som mål å bidra til at digitale tjenester kan brukes av alle, inkludert personer med ulike former for funksjonsnedsettelse.

2. Hva prosjektet går ut på (ta fra rapporten)

Vårt bachelorprosjekt går ut på å utvikle en KI-løsning som skal hjelpe brukerne våre (tjenesteeiere og admins) med å analysere og strukturere tilgjengelighetsfunn fra Testfest til spesifikke GitHub Issues. Testfest samarbeider med Tingtun AS og Norges Blindeforbund, hvor mennesker med ulike funksjonsnedsettelse prøver å teste deres digitale tjenester for å kunne fremme blant annet tilgjengelighetsfunn. En utfordring i dagens prosess er at som oftest er tilbakemeldinger sendt fra testere i ustrukturert form, som for eksempel tekst eller bilder. Som oftest er dette tidkrevende å kunne fikse slik informasjon og omforme den, og derfor er målet med dette prosjektet å finne ut av hvordan man kan bruke KI som et verktøy for å klassifisere tilbakemeldinger. Disse tilbakemeldingene skal blant annet klassifiseres etter WCAG-nivå, ulike temaer, rotårsaker og forslag til eventuelle forbedringer.

3. Utfordringer ved prosjektet

Gruppen møtte på flere utfordringer knyttet til uklar problemforståelse, begrenset forarbeid og innføring av nye teknologier som Google Gemini API og Node.js, noe som førte til usikkerhet og en bratt læringskurv i starten av prosjektet. I tillegg viste det seg krevende å tilpasse arbeidsmetodikk og planlegging underveis, da gruppen gikk fra en ren Scrum-tilnærming til en hybrid waterfall-agil-modell. Dette, kombinert med detaljerte tidsestimater og mindre fleksibel sprintplanlegging, påvirket fremdriften i enkelte perioder. Erfaringene fra prosjektet viser at tydeligere kravavklaring, mer grundig planlegging og mer overordnede estimater kunne bidratt til en mer effektiv gjennomføring.

4. Teknologi / løsninger

Prosjektgruppen utviklet en løsning som analyserer brukertilbakemeldinger ved hjelp av kunstig intelligens og genererer forslag til GitHub Issues. Løsningen er bygget som en fullstack-applikasjon med frontend og backend. Verktøyet ble implementert med bruk av teknologier som:

- JavaScript som språk for både frontend og backend

- React.js for utvikling av brukergrensesnittet
- Node.js for backend-logikk og API-håndtering
- GitHub API for opprettelse og håndtering av issues
- Google Gemini API for KI-basert analyse
- Git og GitHub for versjonskontroll og samarbeid
- Vercel for deploy av frontend
- Render for deploy av backend

Frontend håndterer brukerinput, og backend validerer data og kommuniserer med eksterne tjenester som KI og GitHub API. KI-tjenesten analyserer input og genererer forslag, som gjennomgås og godkjennes før det blir opprettet som en GitHub Issue.

5. Beskrivelse av produkt / resultat

Resultatet av prosjektet er en webbasert løsning som gjør det mulig å registrere, analysere og strukturere tilgjengelighetsfunn ved hjelp av kunstig intelligens. Løsningen er utviklet for å støtte tjenesteeiere og utviklere i arbeidet med å håndtere tilbakemeldinger fra Testfest på en mer effektiv og strukturert måte.

Brukeren kan registrere ett eller flere tilgjengelighetsfunn ved å skrive inn en beskrivelse av problemet og laste opp relevante vedlegg, som bilder eller dokumentasjon. Systemet sender deretter informasjonen til Google Gemini API, som analyserer innholdet og genererer forslag til strukturerte GitHub Issues. Hvert forslag inneholder blant annet problemforståelse, klassifisering etter WCAG-nivå (AA/AAA) og konkrete forslag til utbedringer.

Før GitHub Issues opprettes, får brukeren mulighet til å gjennomgå og redigere hvert funn individuelt. Dette gjør at løsningen fungerer som et støtteverktøy hvor menneskelig kvalitetssikring fortsatt er en sentral del av prosessen. Etter innsending mottar brukeren en bekreftelse med direktelenker til de opprettede GitHub Issues.

Løsningen består av et frontendgrensesnitt utviklet i React.js og en backend utviklet med Node.js. Frontend er designet med fokus på enkel navigering og tydelige brukerflyter, mens backend håndterer kommunikasjon med eksterne API-er og behandling av data.

Bildene illustrerer brukergrensesnittet og flyten i løsningen steg for steg, fra registrering av funn til bekreftelse på opprettede GitHub Issues.

Send inn din rapport

Beskriv problemet eller velg en prompt-mal. Last opp vedlegg og send inn for umiddelbare svar. Se gjennom og send deretter direkte til Github issues.

Konfigurasjon

Unngå å inkludere personopplysninger eller sensitiv informasjon i beskrivelser og vedlegg.

Problem 1

Tittel på sak

Problembeskrivelse*

Vanskelig å se forskjell på fargene

Last opp vedlegg (i form av fil eller bilde)

Last opp 

Har du funnet flere problemer? Trykk på + for å legge til flere problemer.

 Legg til et nytt problem

Send inn

Problem 1

Tittel på sak

Lav fargekontrast og mangelfull visuell koding

Issue Labels

Universell utforming

WCAG-AA

Analyse

1. Klassifisering

Tema: Fargekontrast og visuelle indikatorer

WCAG-nivå: AA

Relevant WCAG-kriterium: 1.4.1 Bruk av farge (A) og 1.4.3 Kontrast (minimum) (AA)

2. Rotårsak

Den underliggende årsaken er at de valgte fargekombinasjonene i grensesnittet har for lav luminansforskjell, noe som resulterer i et kontrastforhold under den lovpålagte kravet på 4.5:1 for vanlig tekst. I tillegg er designet avhengig av farge som eneste kilde til informasjon, uten å bruke sekundære indikatorer som ikoner, mønstre eller tekstlige forklaringer for å skille mellom ulike elementer eller statuser.

3. Konsekvenser for brukere

Brukere med ulike former for fargeblindhet (som protanopi eller deutanopi) vil ikke kunne skille mellom kritiske elementer hvis farge er eneste identifikator. Personer med nedsatt syn eller eldre med redusert kontrastfølsomhet vil oppleve teksten som uleselig. Situasjonshemmede brukere, for eksempel de som bruker skjermen i sterkt sollys, vil også miste evnen til å navigere effektivt.

4. Prioriterte utbedringer

1. Høyeste prioritet: Oppdater fargepaletten i CSS slik at all tekst og tekstbilder har et kontrastforhold på minst 4.5:1 mot bakgrunnen (3:1 for stor tekst over 18pt/24px).

2. Legg til visuelle tilleggsmarkører for å formidle informasjon. Dette innebærer for eksempel å understreke lenker i tekst eller legge til varselikoner med felmeindringer, slik at farge ikke er eneste informasjonsbærer.

3. Sørg for at grafiske komponenter som knapper, skjemafelt og grafer har en kontrast på minst 3:1 mot tilstøtende farger for å sikre synlige skiller.

5. Verifisering

Tilbakene verifiseres ved manuell testing med verktøyet Colour Contrast Analyser (CCA) for å måle nøyaktige kontrastverdier/ Bruk av fargeblindhetssimulatorer i nettlesersens utviklerverktøy (for eksempel i Chrome eller Edge) brukes for å bekrefte at informasjon forblir forståelig i gråtoner eller med simulert fargeblyst. Automatisk tilgjengelighetskontroller som Axe eller Lighthouse kan benyttes for å identifisere gjenværende tekstkontrastell i koden.

Rediger

Send inn

Generert analyse

Problem 1

← Tilbake til alle analyser

Redigerer

Oppdater denne saken

Gjør endringer direkte her. Når du lagrer, oppdateres analysekortet uten at du mister resten av oversikten.

Tittel på sak

Lav fargekontrast og mangelfull visuell koding

Problembeskrivelse

Vanskelig å se forskjell på fargene

Avbryt

Lagre endringer

A screenshot of a web application interface. At the top, a large blue header contains the text "Bekreftelse på innsending" in white. Below this, a white rectangular box with rounded corners contains the text "Dine innsendinger er nå sendt inn til Github Issues som egne saker." followed by "Se de opprettede Github-sakene under, eller start en ny analyse ved å trykke på "Ny sak".". Below this text, there are two buttons: a light blue button labeled "Problem 1" and a dark blue button labeled "Se issue #12". At the bottom of the white box, there is a light blue button labeled "Ny sak/Hjem".

6. Hva vi er spesielt fornøyd med

Gjennom prosjektperioden har gruppen opparbeidet seg verdifull erfaring med både planlegging, utvikling, testing og samarbeid i et større utviklingsprosjekt. Vi er spesielt godt fornøyd med hvordan løsningen gradvis har blitt forbedret gjennom tilbakemeldinger fra oppdragsgiver, veileder og brukere. Prosjektet har gitt oss bedre forståelse for viktigheten av god struktur, tydelig kommunikasjon og jevnlig kvalitetssikring underveis. Vi har også lært hvor viktig det er å dokumentere valg, begrunne avgjørelser og være åpne for justeringer når nye behov eller utfordringer dukker opp. Samlet sett sitter vi igjen med en sterkere forståelse av hvordan et reelt utviklingsprosjekt gjennomføres fra idé og behovskartlegging til ferdig løsning.