IT2 – Prosjekt #2

# Dungeons & Dragons Ability Score Generator

I denne oppgaven skal dere lage en web-app som er en Ability Score- generator til rollespillet Dungeons & Dragons, men som også brukes i andre slike spill.

## Om Ability Scores

I D&D spiller alle som en karakter som har seks forskjellige *egenskaper (ability scores)*, eller stats, som sier noe om karakterens fysiske og mentale evner. Jo høyere score på en egenskap man har, jo sterkere er egenskapen. Egenskapene i D&D er følgende:

* *styrke* (strength)
* *smidighet* (dexterity)
* *konstitusjon* (constitution)
* *intelligens* (intelligence)
* *visdom* (wisdom)
* *karisma* (charisma)

Før man begynner spillet lager man en karakter som har bestemte egenskapsverdier. Det finnes forskjellige måter å avgjøre disse verdiene på, men den mest populære metoden å gjøre det på er å rulle terninger. For å generere egenskapsverdier gjør man følgende:

1. Kast fire seks-sidede terninger (4d6).
2. Ta bort resultatet med lavest verdi.
3. Legg sammen de gjenværende tre resultatene. Dette tallet representerer én egenskapsverdi (tall mellom 3 og 18).
4. Gjenta til man har seks slike verdier.
5. Gi en verdi til hver av de seks egenskapene.

***Eksempel:*** *Lars skal lage sin første D&D karakter og velger å spille som en barbarian. Han begynner med rulle 4 vanlige terninger (4d6). Resultatet er 4, 3, 3, 2. Han fjerner den laveste verdien og legger sammen resten slik at han får 10. Lars gjentar dette fem ganger til og sitter med egenskapsverdiene: 10, 12, 10, 16, 17, 12. Dette er rimelig bra. Siden han spille barbarian vil han fokusere på styrke og konstitusjon så han setter styrken sin til 17 og konstitusjon til 16. Deretter setter han smidighet til 12, intelligens til 10, visdom til 12 og karisma til 10. Nå har Lars definert egenskapene til sin karakter.*

Man kan lese mer på denne lenken hvis man er interessert: <https://www.dndbeyond.com/sources/basic-rules/step-by-step-characters#3DetermineAbilityScores>

## Om oppgaven

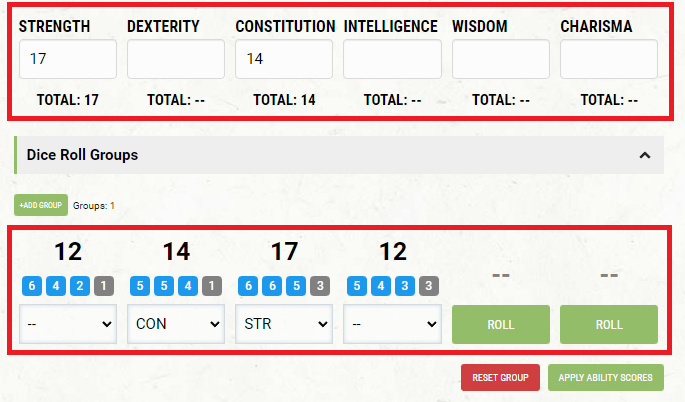
I denne oppgaven skal dere lage en web-app som generer slike egenskapsverdier til en D&D karakter og som følger reglene til spillet som forklart ovenfor. Oppgaven består to deler: implementasjon og testing.

### Implementasjon

Når dere lager web-appen skal designet ta utgangspunkt i bildet under og må oppfylle kravene listet under. Det kan være lurt å tenke både på brukervennlighet og de kravene som oppgaven stiller. Kravene er som følger:

* Må vise hver egenskap med tilhørende egenskapsverdi.
* Må kunne generere seks tilfeldige verdier som skal vises.
* Må kunne tildele egenskaper disse genererte verdiene.

Man kan selvfølgelig legge til andre funksjonaliteter hvis man mener dette vil forbedre programmets brukervennlighet.



*Innholdet i de røde boksene ovenfor er et eksempel på hvordan designet kan se ut, men dere står helt fritt til å velge hvordan det skal se ut så lenge det oppfyller kravene.*

Opprett et SvelteKit-prosjekt som vanlig og programmer de nødvendige funksjonalitetene fra designet ditt. Ettersom man skal gjøre noe likt for seks forskjellige egenskaper er det naturlig her å gjenbruke komponenter i implementasjonen, og man må gjøre dette for å få høy måloppnåelse. I tillegg bør egenskapene implementeres som et **objekt** og genereringen av en tilfeldig egenskapsverdi som en **funksjon**.

### Testing

I den siste delen skal dere skrive en testsamling for web-appen deres. Siden funksjonen(e) dere skal skrive mest sannsynligvis vil gi tilfeldige retur-verdier er det vanskelig å lage gode enhetstester på dette. Derfor må dere skrive en testsamling som kan brukes for å teste grafiske brukergrensesnitt. En slik testsamling skal inneholde testtilfeller som til sammen er dekkende, har en spesifikk fremgangsmåte og et forventet resultat. Denne testsamlingen er litt for deres egen del slik at dere kan oppdage feil eller mangler i web-appen deres og for at jeg kan se at dere kan skrive testtilfeller som er gode og dekkende.

# Levering

Prosjektet skal leveres som en zip-fil på it’sLearning innen **19. november kl. 23:59.** Zip-filen skal inneholde følgende:

* En mappe som inneholder hele SvelteKit-prosjektet.
* Et Word-dokument som inneholder testsamlingen og eventuell beskrivelse og begrunnelse av andre funksjonaliteter dere har lagt til.

# Vurdering

I øktene **mandag 20. og 21. november blir det muntlige høringer** på hver av dere. Dette blir ca. 15-20 minutter der dere gjør rede for prosjektet deres og går igjennom testsamlingen. Prosjektet vil få en karakter vurdert ut fra vurderingskriteriene nedenfor og justerende muntlig høring. Muntlig høringen er mest for å gjøre rede for at arbeidet dere leverer faktisk er deres.

I likhet med tekst skal også kopiert kode kildehenvises til. Kopiert kode fra AI-verktøy bør ikke brukes da det kan være vanskelig å forstå, avvike fra standardene vi bruker og er komplisert å sitere som kilde.

# Vurderingskriterier

|  |  |  |  |
| --- | --- | --- | --- |
| **Kategori** | **Karakteren 2 (bestått)**  **«lav kompetanse»** | **Karakteren 4**  **«god kompetanse»** | **Karakteren 6**  **«utmerket kompetanse»** |
| **Problemløsning** | * forstår og bruker enkle metoder, teknikker og verktøy * beskriver alternative løsninger | * forstår og bruker sammensatte metoder, teknikker og verktøy * forklarer alternative løsninger og tilpasser dem for å oppnå forbedringer | * hensiktsmessig anvende avanserte metoder, teknikker og verktøy * grundig vurderer alternative løsninger og gjøre hensiktsmessige valg |
| **Implementering** | * implementerer enkle løsninger med programmering * bruker hensiktsmessig gjenbrukbar kode * implementerer kode uten syntaksfeil | * implementerer sammensatte løsninger med programmering * bruker og tilpasser hensiktsmessig gjenbrukbar kode * ivaretar vern mot kjøretidsfeil og logiske feil i enkle og sammensatte programmer | * implementerer avanserte løsninger med programmering * bruker, tilpasser og utvikler hensiktsmessig gjenbrukbar kode * ivaretar vern mot kjøretidsfeil og logiske feil i avanserte programmer |