Eksamensoppgave for Avansert Java (PGR203)

- **Emnekode**: PGR203
- **Emnenavn**: Avansert Java
- **Vurderingskombinasjon**: Funksjonalitet og kodekvalitet i fungerende program
- **Eksamensstart**: 2. november
- **Innleveringsfrist**: Tirsdag 10. november 09:00
- **Filformat**: Github classroom-lenke og ZIP-fil fra Github med Java-kildekode og README.md-fil.

Innlevering

- Oppgaven **skal** løses i Github vha Github Classroom med link fra Canvas. Repository på github **skal** være private
- Oppgaven **skal** leveres i *Wiseflow* som en ZIP-fil og link til Github Classroom
- Oppgaven **skal** løses parvis. Ønsker du å levere alene eller i gruppe på tre må dette avklares med foreleser innen siste forelesning
- Innleveringen kan deles med en annen gruppe for gjensidig tilbakemelding. Tilbakemelding skal gis i form av Github issues
- README.md på Github **skal** linke til Github Actions som skal kjøre enhetstester uten feil. README-filen skal også inneholde link til gitt tilbakemelding til annet team, et UML-diagram samt beskrivelse av hva kandiditene ønskes skal vurderes i evalueringen av innleveringen
- Koden **skal** lese database settings fra en fil som heter `pgr203.properties` og ser ut som følger:

```
dataSource.url=...
dataSource.username=...
dataSource.password=...
```

Oppgave

Mappeoppgaven for PGR203 er å lage en backend server for å håndtere prosjektstyring med tildeling og status på oppgaver. Serveren skal la bruker kunne opprette oppgaver, redigere oppgavene, tildele oppgaver til prosjektdeltagere og endre oppgavestatus.

Programmet skal utvikles på en måte som demonstrerer programmeringsferdigheter slik det vises i undervisningen. Spesielt skal all funksjonalitet ha automatiske tester og være fri for grunnleggende sikkerhetssvakheter. Programmet skal demonstre at kandidatene mestrer Sockets og JDBC bibliotekene i Java.

Programmet skal leveres i form av et maven prosjekt som kan bygge en `executable jar` og lagre data i en PostgreSQL-database. Oppgaven skal leveres i grupper på to.

Dere skal levere en webserver som skal kunne brukes i nettleseren for å administrer oppgaver i et prosjektstyringssystem.

Funksjonaliteten skal inkludere:

- Opprett prosjektmedlem med navn og email
- Liste prosjektmedlemmer
- Opprett ny prosjektoppgave med navn og status
- Tildel oppgave til prosjektmedlemmer
- Liste opp prosjektoppgaver, inkludert status og tildelte prosjektmedlemmer
- Endre oppgavestatus
- Filtrere oppgaver per prosjektmedlem og status

Ekstra funksjonalitet gir bonuspoeng. Dette kan for eksempel være redigering av eksisterende prosjektmedlemmer og oppgaver, organisering av oppgaver og prosjektmedlemmer i prosjekt, "logge inn" som prosjektmedlem med cookies, endre statuskategorier.

Prosjektet må følge god programmeringsskikk. Dette er viktige krav og feil på et enkelt punkt kan gi en hel karakter i trekk.

- Koden skal følge standard Java-konvensjoner med hensyn til store og små bokstaver og indentering
- Koden skal utvikles på Git, med Maven og kjøre tester på Github Actions
- Koden skal ha god testdekning
- Det skal ikke forekomme SQL Injection feil
- Databasepassord skal være konfigurert i en fil som *ikke* sjekkes inn i git
- README-fil må dokumentere hvordan man bygger, konfigurerer og kjører løsningen
- README-fil må dokumentere designet på løsningen
- README-fil må beskrive erfaringene med arbeidet og løsningen
- Maven skal bygge en executable jar som inneholder HTML-koden og som referer til `pgr203.properties` i current working directory

God abstraksjon for eksempel DAO eller Controller-klasser gir bonuspoeng. Konsistent bruk av begreper og kode vektlegges.

Brukervennlighet *er **ikke*** et vurderingskriterie for karakteren.

Vedlegg: Eksempel på frontend kode

`index.html`

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>KristianiaProject</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
<h1>Project view</h1>

<div id="projectMembers"></div>

<div>
    <a href="newWorker.html">Add new project member</a>
    <a href="listTasks.html">Vis oppgaver</a>
</div>
</body>
<script>
fetch("/api/projectMembers")
    .then(function(response) {
        return response.text();
    }).then(function(text) {
        document.getElementById("projectMembers").innerHTML = text;
    });
</script>
</html>
```

newWorker.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Add project member | KristianiaProject</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
<h1>Add project member</h1>

<form method="POST" action="/api/members">
    <label>Member name: <input type="text" name="full_name" /></label>
    <label>Member email: <input type="text" name="email" /></label>
    <button>Add</button>
</form>

<div>
    <a href=".">Return to front page</a>
</div>
</body>
</html>
```

newWorker.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>List tasks | KristianiaProject</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
<h1>Add project member</h1>

<form method="GET" action="/api/tasks">
    <label>Member name: <select id="projectMembers" name="projectMember" /></label>
    <!-- TODO: Filter på status -->
    <button>List</button>
</form>

<div>
    <a href=".">Return to front page</a>
</div>
</body>
<script>
fetch("/api/projectMembers")
    .then(function(response) {
        return response.text();
    }).then(function(text) {
        document.getElementById("projectMembers").innerHTML = text;
    });
</script>
</html>
```

`style.css`

```
body {
    background-color: lime;
}
```

Vedlegg: Sjekkliste for innlevering

- [ ] Dere har lastet opp en ZIP-fil med navn basert på navnet på deres Github repository
- [ ] Koden er sjekket inn på github.com/kristiania-repository
- [ ] Dere har committed kode med begge prosjektdeltagernes GitHub konto (alternativt: README beskriver arbeidsform)

README.md

- [ ] `README.md` inneholder en korrekt link til Github Actions
- [ ] `README.md` beskriver prosjektets funksjonalitet, hvordan man bygger det og hvordan man kjører det
- [ ] `README.md` beskriver eventuell ekstra leveranse utover minimum
- [ ] `README.md` inneholder et diagram som viser datamodellen
- [ ] Dere har gitt minst 2 positive og 2 korrektive GitHub issues til en annen gruppe og inkluderer link til disse fra README.md

Koden

- [ ] `mvn package` bygger en executable jar-fil
- [ ] Koden inneholder et godt sett med tester
- [ ] `java -jar target/...jar` (etter `mvn package`) lar bruker legge til og liste ut data fra databasen via webgrensesnitt
- [ ] Programmet leser `dataSource.url`, `dataSource.username` og `dataSource.password` fra `pgr203.properties` for å connecte til databasen
- [ ] Programmet bruker Flywaydb for å sette opp databaseskjema
- [ ] Server skriver nyttige loggmeldinger, inkludert informasjon om hvilken URL den kjører på ved oppstart

Funksjonalitet

- [ ] Programmet kan liste prosjektdeltagere fra databasen
- [ ] Programmet lar bruker opprette nye prosjektdeltagere i databasen
- [ ] Programmet kan opprette og liste prosjektoppgaver fra databasen
- [ ] Programmet lar bruker tildele prosjektdeltagere til oppgaver
- [ ] Flere prosjektdeltagere kan tildeles til samme oppgave
- [ ] Programmet lar bruker endre status på en oppgave

Vedlegg: Mulighet for ekstrapoeng

- [ ] Håndtering av request target "/"
- [ ] Avansert datamodell (mer enn 3 tabeller)
- [ ] Avansert funksjonalitet (redigering av prosjektmedlemmer, statuskategorier, prosjekter)
- [ ] Implementasjon av cookies for å konstruere sesjoner: https://tools.ietf.org/html/rfc6265#section-3
- [ ] UML diagram som dokumenterer datamodell og/eller arkitektur (presentert i README.md)
- [ ] Rammeverk rundt Http-håndtering (en god HttpMessage class med HttpRequest og HttpResponse subtyper) som gjenspeiler RFC7230
- [ ] God bruk av DAO-pattern
- [ ] God bruk av Controller-pattern
- [ ] Korrekt håndtering av norske tegn i HTTP
- [ ] Link til video med god demonstrasjon av ping-pong programmering
- [ ] Automatisk rapportering av testdekning i Github Actions
- [ ] Implementasjon av Chunked Transfer Encoding: https://tools.ietf.org/html/rfc7230#section-4.1
- [ ] Annet