

# Slutprojekt

## Grit Academy Portal

I det här projektet kommer vi att skapa en webbportal. Vi kommer att använda Java-miljön för att uppnå detta mål. Grit Academy Portal är ett centralt nav för studenter, lärare och administratörer för att hantera Grit Academy-resurser. Tre delar i detta projekt är att skapa en databas, skapa webbsidor, skapa användarhierarki för hantering av resurser. I det här projektet kommer vi att använda MySQL och phpMyAdmin för att skapa databasen och Java Servlet, JSP, HTML, CSS för att skapa webbsidor. Interaktionen med den här portalen baseras på vilken typ av användare som är ansluten.

Skapa Github repo med din kod. Länka koden i Google Classroom. Skriv max 1 A4-sida rapport om vad som var intressant, utmanande etc. Namnge repot såhär: **Java23-WebDevJava-GritAcademyPortal-fornamn-eternamn**

### G – Krav

#### Användare

- Användare av "Grit Academy Portal" kategoriseras som en av 2 *user type*. Dessa typer är:
  - **student** – Denna typ används för alla elever på skolan
  - **teacher** – Denna typ används för alla lärare på skolan
- Varje *user type* har en viss uppsättning privilegier baserat på en av 2 *privilage type* som den har. *Privilage type* avgör vilka databasåtgärder som kan utföras av användare av "Grit Academy Portal". *Privilage type* kan vara något av följande:
  - **user** – Det här är en standardtyp för alla *user type*. Den här *privilage type* har READ privilegium i databasen.
    - Om en *user type* är **student** kan den se alla kurser som den deltar i med alla studenter och lärare i dessa kurser.
    - Om en *user type* är **teacher** kan den se:
      - Alla kurser

- Alla elever
- **admin** – Den här *privilage type* har **CRUD**-privilegier för alla tabeller i databasen.
  - Den här *privilage type* kan se allt som *privilage type user* kan se
  - Endast en *user type teacher* kan upphöjas till en *privilage type admin*.
  - Endast *privilage type admin* kan registrera lärare och elever och lägga till nya kurser.
  - Endast *privilage type admin* kan registrera associationer mellan lärare och kurser.
  - Endast *privilage type admin* kan registrera associationer mellan elever och kurser.
  - Endast *privilage type admin* kan ge annan lärare *privilage type admin*
  - Det bör finnas en *user type teacher* med *privilage type admin* i tabellen **Teachers** från början.
- Varje användare har ett av 2 *state type*:
  - **anonymous** – detta *state type* är ett standardtillstånd för alla som ansluter till "Grit Academy Portal"
  - **confirmed** – Det här *state type* gäller för alla *user type* som är inloggade
- För att logga in måste en *user type* registreras som **student** eller som **teacher**. INTE BÅDA
- För att kunna utföra operationer i databasen måste en "Grit Academy Portal"-*user type* vara inloggad alltså måste ha *state type confirmed*.

**Databas:** Databas och tabeller kan skapas med phpMyAdmin.

- Skapa en databas **GritAcademy**
- Databasen bör ha tabeller med kolumner som definieras nedan:
  - **Students** (tabell för students)
    - **id** (PK)
    - **fName**
    - **lName**
    - **town**
    - **email**
    - **phone**
    - **username**
    - **password**
  - **Courses** (tabell för courses)
    - **id** (PK)
    - **name**
    - **YHP**
    - **description**
  - **Teachers** (tabell for teachers)

- **id** (PK)
- **fName**
- **lName**
- **town**
- **email**
- **phone**
- **username**
- **password**
- **privilage\_type** (kan vara user eller admin)
- **Students\_Courses** (tabell för association mellan **Students** och **Courses**)
  - **id** (PK)
  - **students\_id** (FK Students.id)
  - **courses\_id** (FK Curses.id)
- **Teachers\_Courses** (tabell för association mellan **Teachers** och **Courses**)
  - **id** (PK)
  - **teachers\_id** (FK Teachers.id)
  - **courses\_id** (FK Courses.id)

**Webbsidor:** Alla webbsidor ska skapas med hjälp av: Servlets, JSP, HTML, och CSS. Alla webbsidor ska ha Navbar, och Footer.

- **Index**
  - En hemsida som beskriver "Grit Academy PORTAL"
    - Alla *state type* kan se den här sidan
- **Courses**
  - En webbsida som visar alla tillgängliga kurser i skolan
    - Alla *state type* kan se den här sidan
- **Login**
  - En webbsida med ett inloggningsformulär
    - Alla *state type* kan se den här sidan
    - Inloggad användare får *state type* **confirmed**
    - Användare kan logga in med **username** och **password** om *state type* är **anonymous**
    - Det bör finnas en utloggningsknapp för användare med *state type* **confirmed**
- **Register**
  - En webbsida för registrering av studenter och lärare
    - Bara *user type* **teacher** med *privilage type* **admin** och *state type* **confirmed** kan se den här sidan
    - Rätt *user type* ska tilldelas vid registrering
- **UserPage**
  - En hemsida för alla *user type*

- Alla *user type* med *state type* **confirmed** kan se den här sidan
- Den här sidan bör tillåta *user typ* att utföra åtgärder baserat på *user typ* och den *privilage type* de har.

## VG

Istället för att ha en *user type* **teacher** med *privilage type* **admin** från början, skapa ytterligare en *privilage type* **superadmin**. Endast *privilage type* **superadmin** ska kunna bevilja *privilage type* **admin** till *user type* **teacher**.

- Skapa statistik wepage
  - Endast *user type* **teacher** med *privilage type* **superadmin** och *state type* **confirmed** ska kunna se detta
  - Statistik bör visa:
    - Genomsnittligt antal kurser som tagits av elever.
    - Kurser i popularitetsordning
    - Studenter i ordning efter de flesta kurser
  - Om en *user type* är **teacher** kan den se:
    - Alla kurser
    - Alla elever
    - Alla kurser för en specifik elev
    - Alla elever och lärare för en specifik kurs
    - INTE BARA DE HEN ÄR INVOLVERAD !!

## Tips:

- Börja med att förstå alla användartillstånd
- *user type* ska sättas i rätt tabell och kan hanteras med Java Bean och Context eller Session
- *privilage type* ska sättas i rätt tabell och kan hanteras med Java Bean och Context eller Session
- *user type* **student** har *privilage type* **user** *alltid*
- *user type* **teacher** börjar med *privilage type* **user**
- *state type* kan hanteras med Java Bean och Session. Ställ in det på rätt värde efter lyckad inloggning
- *state type* är *alltid* **anonymous** Innan någon loggar in
- Skapa JSP-fragment för Navbar och Footer
- Använd en CSS-fil för formatering.

- Skapa Servlet för alla webbsidor och vidarebefordra användaren till rätt JSP-sida baserat på *user type*, *privilage type* och *state type*.
- Använd Context för globala variabler och data.
- Använd Java Bean som representerar användare
- Glöm inte MySQL och JSTL dependency och tomcat plugin in POM.xml

**Lycka till!**

[Lukas.kurasinski@gritacademy.se](mailto:Lukas.kurasinski@gritacademy.se)