

# Blog App User Roles

## Blog Application with User Roles and Authentication

### Project Overview

This project aims to develop a web application for managing blog posts with user roles and authentication. The application will have two user roles: Admin and Viewer.

### Technologies

- **Backend:** Java Servlets
- **Frontend:** JSP (JavaServer Pages)
- **Database:** MySQL

### Project Phases

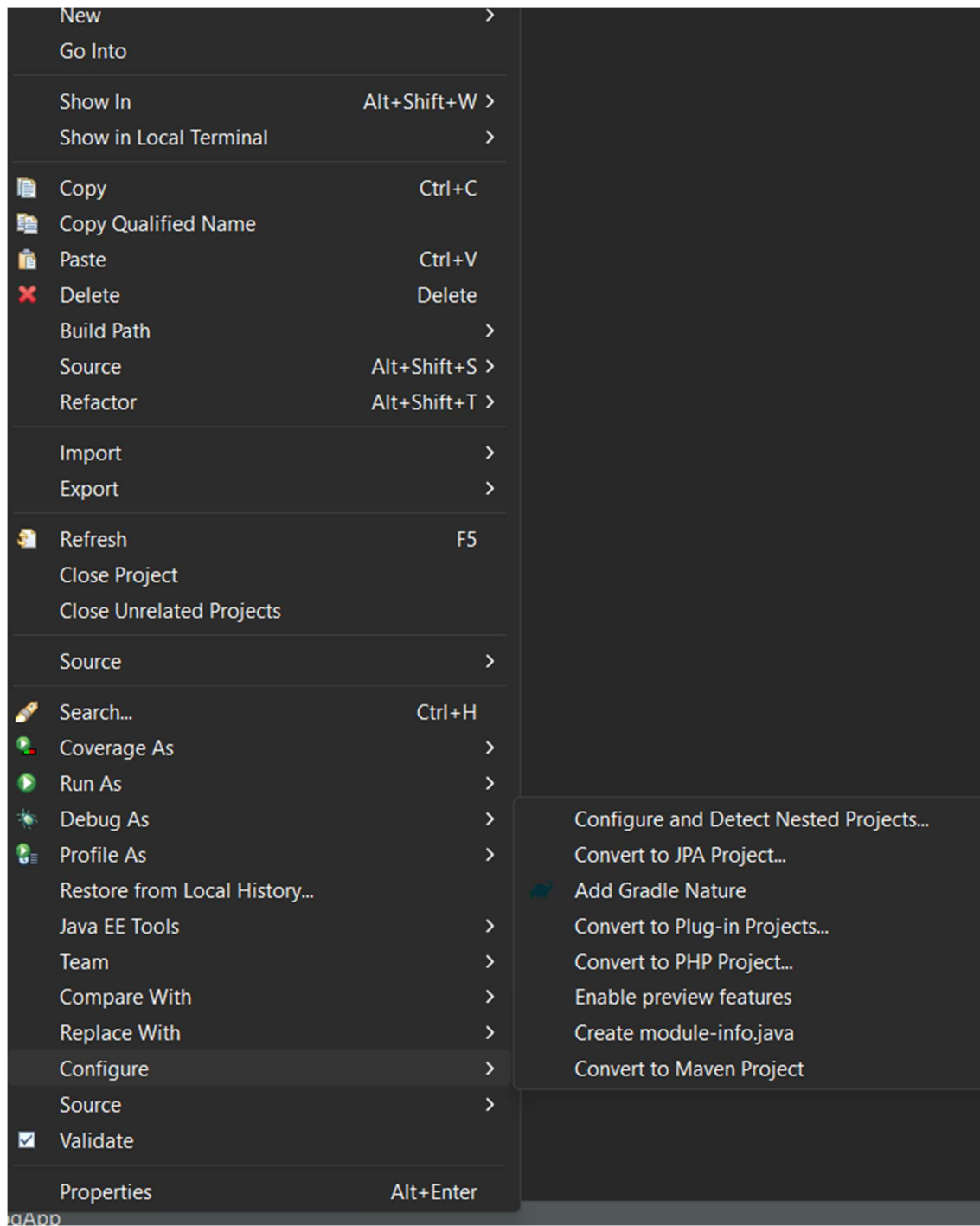
#### Creating java project

#### Follow some steps:

Create Java Dynamic Web Project

Click on project

It will display



Click on configure and convert to maven project

In pom.xml it will shows and add dependencies

```
<project
xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-
4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>BlogApplication1</groupId>

  <artifactId>BlogApplication1</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>war</packaging>
  <build>
    <plugins>
      <plugin>
        <artifactId>maven-compiler-
plugin</artifactId>
        <version>3.8.1</version>
        <configuration>
          <release>17</release>
        </configuration>
      </plugin>
      <plugin>
        <artifactId>maven-war-
plugin</artifactId>
        <version>3.2.3</version>
      </plugin>
    </plugins>
  </build>

  <dependencies>
    <dependency>
```

```
        <groupId>org.mindrot</groupId>
        <artifactId>jbcrypt</artifactId>
        <version>0.4</version>
    </dependency>

    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-
java</artifactId>
        <version>8.0.30</version>
    </dependency>

</dependencies>

</project>
```

Add Dependencies

```
<dependencies>
    <dependency>
        <groupId>org.mindrot</groupId>
        <artifactId>jbcrypt</artifactId>
        <version>0.4</version>
    </dependency>
```

```
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-
java</artifactId>
            <version>8.0.30</version>
        </dependency>

    </dependencies>
```

### Create tables

**CREATE TABLE user (**

**id INT AUTO\_INCREMENT PRIMARY KEY,**

**name VARCHAR(100) NOT NULL,**

**email VARCHAR(100) NOT NULL UNIQUE,**

**password VARCHAR(255) NOT NULL,**

**role ENUM('Admin', 'Viewer') NOT NULL**

**);**

The screenshot shows a database management interface with a toolbar at the top. Below the toolbar, a SQL query is entered in a text area: `SELECT * FROM blogapp.user;`. Below the query area, a 'Result Grid' displays the results of the query. The grid has columns for 'id', 'name', 'email', 'password', and 'role'. There are four rows of data shown.

| id | name  | email           | password                                     | role   |
|----|-------|-----------------|--|--------|
| 13 | ram   | ram@gmail.com   | \$2a\$10\$ZLThdo04A6nZUa5EWYNVO6LcGPa4e...   | Admin  |
| 14 | Akhil | akil@gmail.com  | \$2a\$10\$uifMx5dNzXwuANTjOkWMOjDSwR3ZGi...  | Admin  |
| 15 | Rahul | rahul@gmail.com | \$2a\$10\$bWkehMegYTRL/jx8Jqsv1uGGMUZdN61... | Viewer |
| 18 | John  | john@gmail.com  | \$2a\$10\$87PFHvIgQW17/OREXQA/.KmNN6wK3...   | Admin  |

```
CREATE TABLE posts (
  id INT AUTO_INCREMENT PRIMARY KEY,
  title VARCHAR(255),
  content TEXT,
  date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  userId INT,
  edit BOOLEAN DEFAULT false,
  FOREIGN KEY (userId) REFERENCES user(id)
);
```

```
CREATE TABLE media (
  id INT AUTO_INCREMENT PRIMARY KEY,
  postId INT,
  file_data LONGBLOB,
  upload_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (postId) REFERENCES posts(id) ON DELETE CASCADE
```

);

**Sample details:**

**User name :** Ram

Email : [ram@gmail.com](mailto:ram@gmail.com)

Password : ram123

Role: Admin

User name : Akil

Email : [akil@gmail.com](mailto:akil@gmail.com)

Password : 12345678

Role: Admin

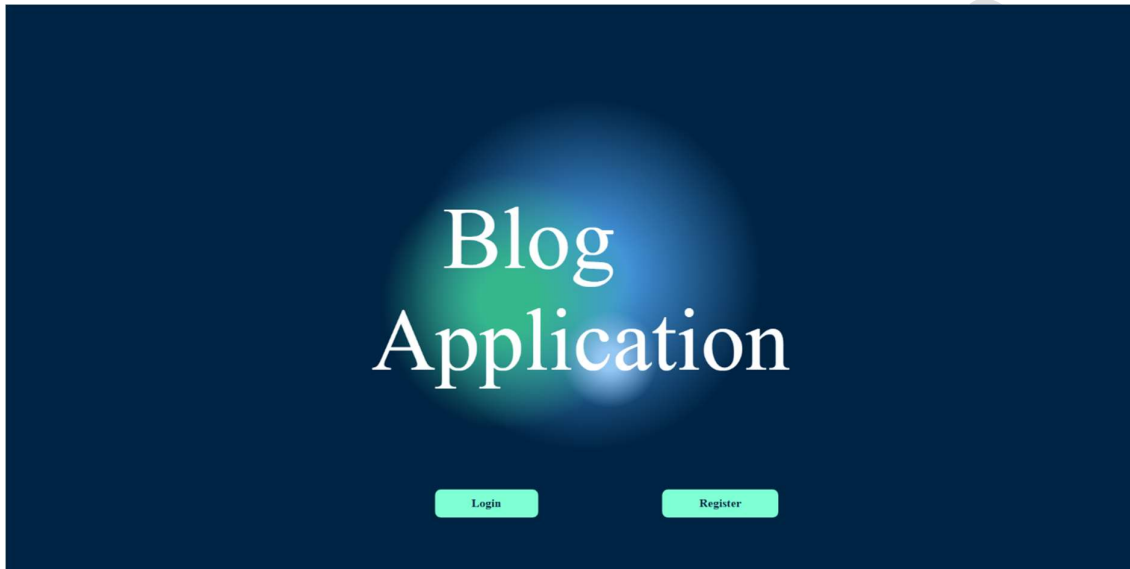
User name : Rahul

Email : [rahul@gmail.com](mailto:rahul@gmail.com)

Password : rahul1234

Role: viewer

Index.jsp




## Phase 1: User Registration and Login (Frontend & Backend)

### 1. User Registration

- Create a user registration form that captures user details (name, email, password) and role (Admin/Viewer).
- Implement user registration logic in a Servlet:
  - Validate user data.
  - Store user data in the MySQL database.
  - Assign roles (Admin/Viewer).



A screenshot of a web application's registration form. The form is titled "Register" and is centered on a dark blue background. It contains four input fields: "User Name", "Email", and "Password", each with a label above it. Below these is a "Role" dropdown menu currently showing "Admin". At the bottom is a "Register" button.

**Register**

User Name

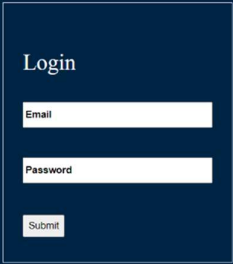
Email

Password

Role :

## 2. User Login

- Create a login page with username/email and password fields.
- Implement login logic in a Servlet:
  - Authenticate users against the database.
  - Upon successful login, establish user sessions using cookies or HTTP sessions for role identification.

A screenshot of a web application's login form. The form is titled "Login" and is centered on a dark blue background. It contains two input fields: "Email" and "Password", each with a label above it. Below these is a "Submit" button.

**Login**

Email

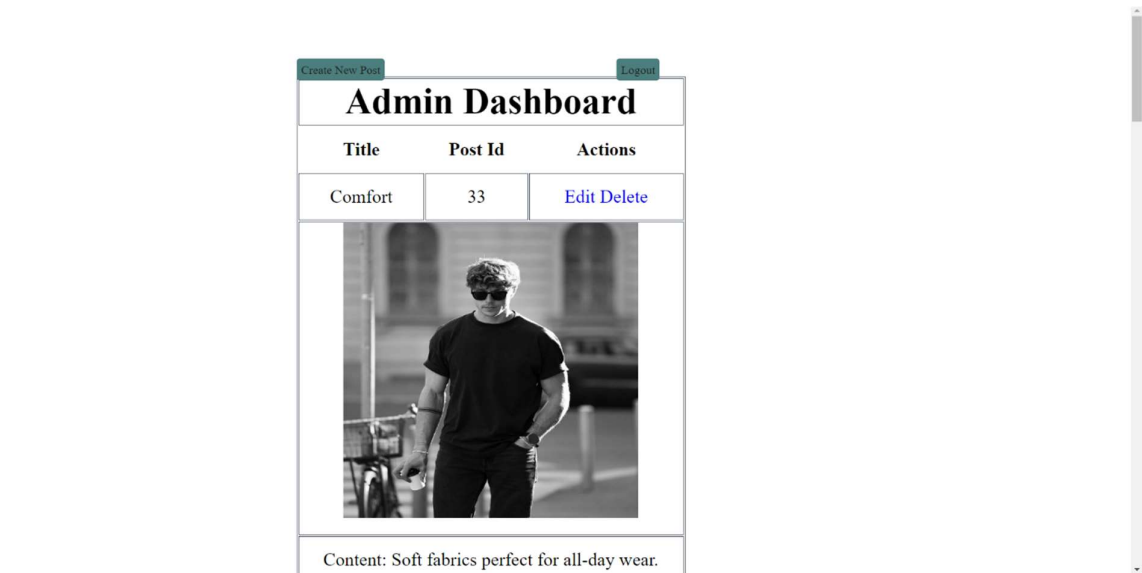
Password

## Phase 2: Admin Dashboard (Frontend & Backend)

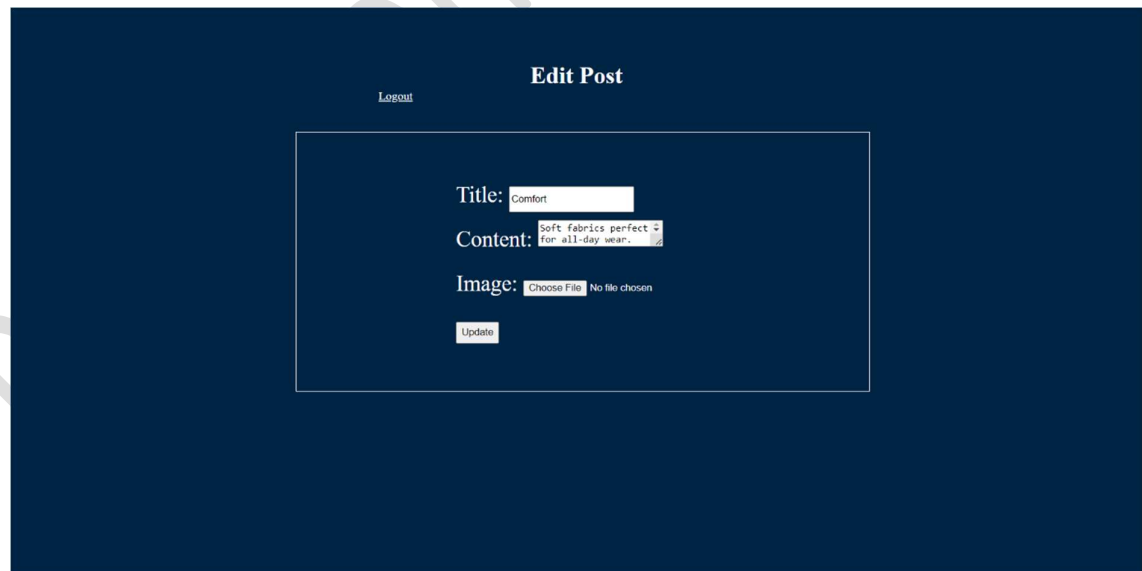
### 1. Admin Dashboard

- Develop an Admin dashboard accessible only after successful login with the "Admin" role.
- Implement functionalities for Admin to:

- Create new blog posts with title, content, and an option to upload images.
- Update existing blog posts (title, content, image).
- Delete blog posts.



Edit pOst



### Phase 3: User Interface for Viewers (Frontend)


#### 1. User Interface for Viewers

- Search blog posts by title or date.(still not implement)

- View individual blog posts in detail, opening them in a new tab.

| Viewer Dashboard |                      |
|------------------|----------------------|
| Admin            | Action               |
| John Doe         | <a href="#">View</a> |
| ram              | <a href="#">View</a> |
| ramesh           | <a href="#">View</a> |

Openend Admin John Doe in new tab

| Admin Posts   |         |    |
|---|---------|----|
| Search by Title: <input type="text"/>   |         |    |
| <input type="button" value="Search"/>   |         |    |
| Title   | Post Id |    |
|   | Clarity | 39 |
|  |         |    |

Video Link::

<https://1drv.ms/v/s!Avd4fnSa-w6DgW1mPQPc4SLCoWBM>

Or

<https://photos.onedrive.com/share/830EFB9A747E78F7!237?cid=830EFB9A747E78F7&resId=830EFB9A747E78F7!237&authkey=!AGY9A9zhIsKhYEw&ithint=video>

---

Purushotham Palla